

TARTU ÜLIKOOL  
LOODUS- JA TEHNOLOOGIATEADUSKOND  
Füüsika instituut  
Arvutitehnika eriala

**Arvi Kiik**  
**LEGO Mindstorms NXT roboti**  
**programmeerimine keeles NXC**  
**(e-kursuse loomine)**  
**Magistritöö (30 EAP)**

Juhendajad: Anne Villems  
Taavi Duvin

Autor: ..... “.....” mai 2013  
Juhendaja: ..... “.....” mai 2013  
Juhendaja: ..... “.....” mai 2013

Lubada kaitsmisele

Professor ..... “.....” mai 2013

TARTU 2013

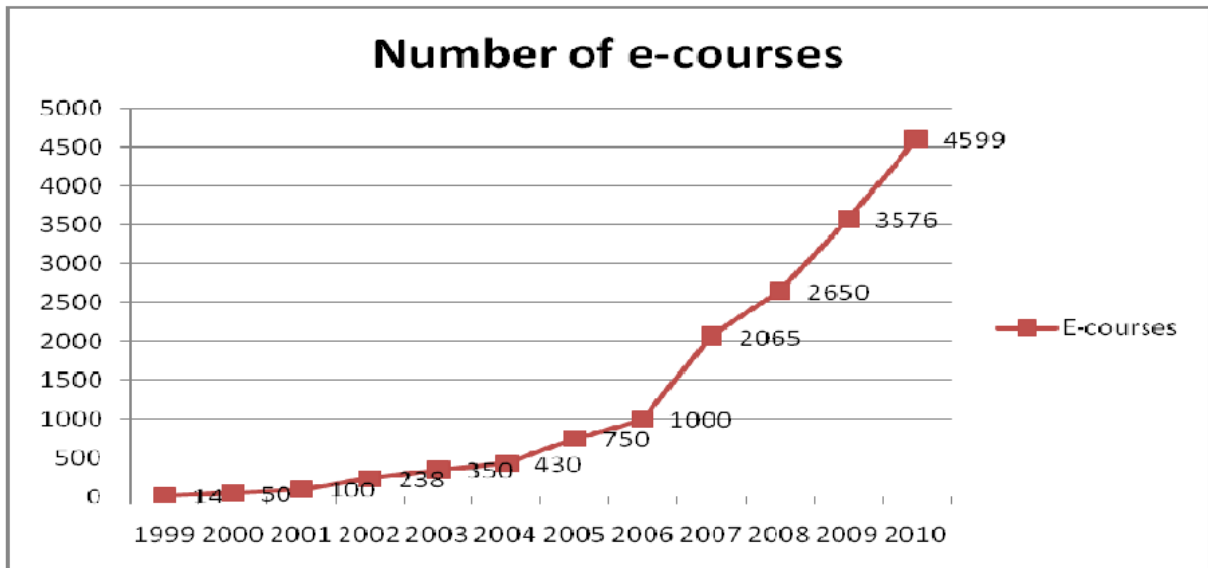
# Sisukord

Sisukord .....	2
Sissejuhatus .....	4
1. Olemasolevad keeled LEGO Mindstorms NXT programmeerimiseks .....	7
1.1. Üleminek teisele programmeerimiskeelele .....	10
1.2. NXC keel teise programmeerimiskeelena .....	12
1.3. NXC keele tutvustus .....	17
1.4. NXC keele süntaksist .....	19
1.5. NXT-G ja NXC keelest baitkoodi teisendatud programmide mälu kasutuse võrdlus .....	22
1.6. NXT-G ja NXC keeles kirjutatud programmide täitmise kiiruse võrdlus .....	25
2. NXCEesti .....	30
2.1. NXCEesti installeerimisjuhend .....	30
2.2. NXCEesti kasutajaliides .....	32
2.3 NXCEesti eestikeelsete funktsioonide teek .....	33
2.4 NXCEesti näidisprogramm .....	36
3. NXC e-kursuse loomise protsess .....	39
3.1 E-kursuse analüüs .....	40
3.2 Õppeprotsessi kavandamine .....	44
3.3 Kursuse loomine .....	48
3.4 Test baasteadmsite peale .....	54
3.5 Kursuse haldamine .....	55
Kokkuvõte .....	59
LEGO Mindstorms NXT robotics programming in NXC language .....	61
Kasutatud kirjandus .....	63

Lisad.....	67
Lisa 1 .....	67
Lisa 2 .....	67
Lisa 3 .....	67

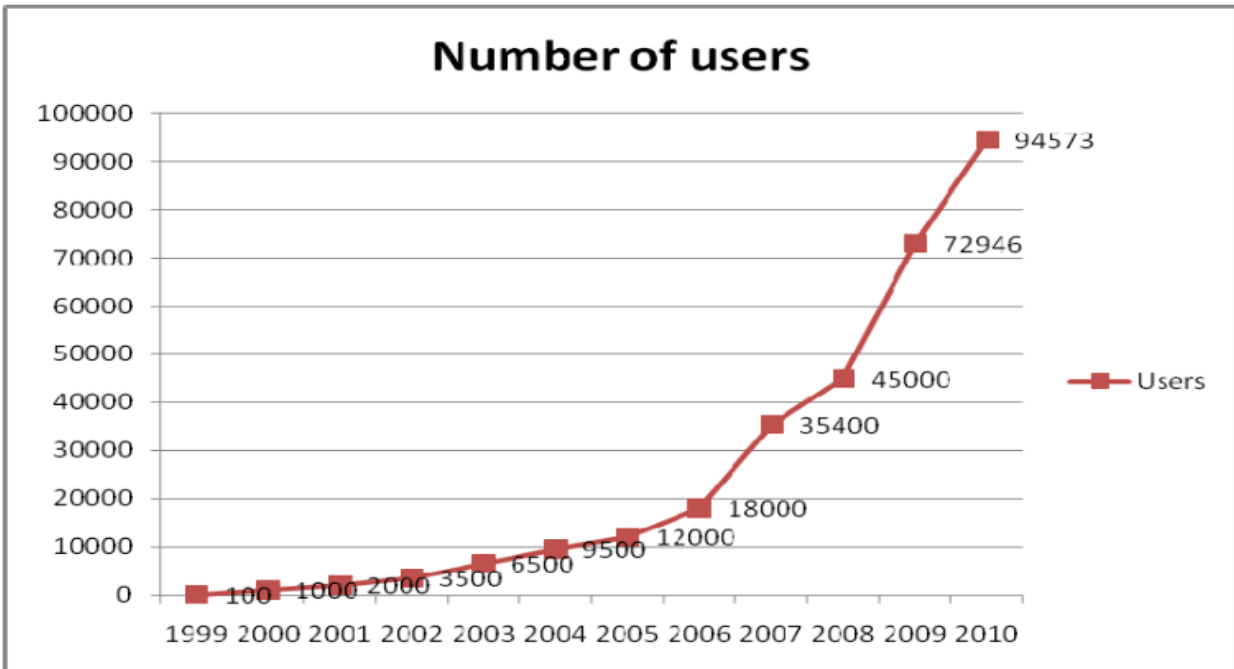
## Sissejuhatus

E-õpe on moderne ja efektiivne õppimise ja õpetamise meetod, mis on muutunud järjest populaarsemaks nii üldharidus- kui ka ülikoolides. E-kursuste arv on alates 1999 aastast Eestis kasvanud märkimisväärselt ( vt joonis 1 ).



**Joonis 1** E-kursuste arvu kasv e-Õppe Arenduskeskuse poolt hallatavates õpikeskkondades Moodle ja Blackboard ajavahemikul 1999-2010 [1]

Jooniselt 1 on näha, et e-kursuste arv e-Õppe Arenduskeskuse poolt hallatavates õpikeskkondades on 2010 aasta lõpu seisuga peaaegu 4600. Samuti on suurenenud ka kursusest osavõtjate arv, mis küünib ligikaudu 94600-ni ( vt joonis 2 ).



**Joonis 2** E-kursusest osalejate arvu kasv e-Õppe Arenduskeskuse poolt hallatavates õpikeskkondades Moodle ja Blackboard ajavahemikul 1999-2010 [1]

Antud magistritöös püstitatud ülesandeks on luua olemasoleva LEGO Mindstorms NXT algkursuse kõrvale uus kursus nimega LEGO Mindstorms NXT roboti programmeerimine keeles NXC. E-kursus peab olema korraga kasutatav mitme huvigrupi poolt (õpilased ja õppivaid õpetajad). Töö raames selgitatakse välja täpsed nõuded uuele kursusele. E-kursuse loomiseks kasutatakse Moodle e-õppe keskkonda ja sealseid arendusvahendeid.

Antud magistritöö eesmärgid:

1. Põhjendada uuele keelele ülemineku vajalikkust
2. Anda ülevaade NXC keelest
3. LEGO Mindstorms roboti programmeerimine keeles NXC
4. NXCEesti kasutusjuhend
5. Määrata täpsed nõuded uuele e-kursusele
6. Kirjeldada kogu kursuse loomise protsessi
7. Programmeerimisülesannete loomine

Magistritöö koosneb kolmest osast. Esimeses osas käsitletakse lahendatavat probleemi täpsemalt ja tutvustatakse NXC keelt. Teine osa tutvustab Priit Ranna loodud programmi NXCEesti ja selle kasutamist ning võimalusi, kuidas sellega NXC keeles programme kirjutada. Kolmas osa kirjeldab uue kursuse loomise protsessi ning antakse ülevaade loodud kursusest.

Kursus peab olema toeks RAJU keskuse poolt korraldavatele õpetajakoolitustel osalevatele robotika õpetajatele ning võimaldama ka robotika alaseid lõputöid kirjutatavatel tudengitel saada baasteadmisi NXC keelest.

# 1. Olemasolevad keeled LEGO Mindstorms NXT programmeerimiseks

LEGO Mindstorms NXT on LEGO väljatöötatud robotikakomplekt, mis tuli välja aastal 2006. See annab õpilasele hea võimaluse ehitada robot ja seda erinevate ülesannete täitmiseks programmeerida arendades seeläbi programmeerimisoskust ning loogilist mõtlemist. Roboti programmeerimiseks on võimalik kasutada erinevaid programmeerimiskeeli. Tuntumad programmeerimiskeeled, mida NXT juhtploki programmeerimiseks kasutatakse, on NXT-G, NXC, RobotC, ROBOLAB, LABview, leJOS NXJ, NXT\_Python ja NBC. Järgnevas osas antaksegi lühiülevaade nendest keeltest [2].

Põhiline programmeerimiskeel, mida NXT roboti programmeerimiseks kasutatakse, on **NXT-G** keel. NXT-G keel on kõige populaarsem, kuna see keel on LEGO enda poolt selle roboti programmeerimiseks loodud ning seda on võimalik LEGO Mindstorms kodulehelt alla laadida. Tarkvara on tasuta [3]. Seda esimese programmeerimiskeelena lihtne kasutama õppida, kuna tegu on lastele (alates 8.eluaastast) mõeldud graafilise keelega.

Programmeerimine selles graafilises keskkonnas toimub plokkide vedamise ja nende omavahelise ühendamise teel. Graafiliselt kujult kompileeritakse programmi kood baitkoodi, mida seejärel interpreteeritakse juba NXT juhtplokis oleva tarkvaraga. NXT-G keele miinusteks on programmide suur maht baitkoodis, täitmise suhteliselt pikk aeg ning keelel puudub ametlik tugi massiivide kasutamiseks.

**NXC** on C keele sarnane programmeerimiskeel, mis on loodud John Hanseni poolt LEGO Mindstorms NXT robotite programmeerimiseks [4]. Parimaks NXC keeles programmide loomise arenduskeskkonnaks Windows operatsioonisüsteemides on Brixx Command Center [5], Mac OS süsteemide puhul NeXT Command Center [6]. Linux platvormile puudub spetsiaalselt NXC kasutamiseks mõeldud vahend. Järgnevalt on toodud näide NXC programmi koodist ( vt näide 1 ).

```

task main()
{
    OnFwd(OUT_A, 75);
    OnFwd(OUT_C, 75);
    Wait(4000);
    OnRev(OUT_AC, 75);
    Wait(4000);
    Off(OUT_AC);
}

```

**Näide 1** Programmist, kus robot sõidab 4 sekundit edasi, seejärel 4 sekundit tagasi ning lülitab siis mootorid A ja C välja [7]

Antud näites pannakse mootorid, mis on ühendatud porti A ja C, *OnFwd* funktsiooniga edasi sõitma. Tegevust tehakse 4 sekundit, seejärel *OnRev* funktsiooniga pannakse robot tagurdama. Lõpus lülitatakse mootorid *Off* funktsiooni kasutades välja.

**RobotC** on programmeerimiskeel, mis põhineb traditsioonilisel C keelel. Programmeerimine selles keskkonnas toimub sarnaselt NXT-G keskkonnale, kus lohistatakse erinevaid plokkide omavahel kokku. Vahel on selles, et graafiliste plokkide asemel on koodiplokkid teksti kujul. Eeliseks on reaajas töötav silur (ing. k. *real-time debugger*), mis vähendab tunduvalt programmide silumiseks kuluvat aega. Miinuseks on see, et antud tarkvara on tasuline. Lisaks on ROBOTC-l olemas kaks erinevat taset: tavatase ja eksperttase. Tavatasemel on paljud spetsiifilised eksperttasemel nähtavad funktsioonid peidetud [8].

**ROBOLAB** on originaalis Tuftsi Ülikooli poolt LEGO Mindstorms RXC (NXT-le eelnenud põlvkond) jaoks valmistatud graafiline programmeerimiskeskond. Hiljem programmi sisse viidud muudatused võimaldavad seda kasutada ka NXT robotite programmeerimisel. Sarnaselt NXT-G-le toimub ka selles aplikatsioonis programmeerimine ikoonide abil [9].

**LABview** on National Instrumentsi poolt arendatud graafiline programmeerimiskeskond, millega on võimalik programmeerida ka LEGO NXT roboteid. Programmeerimine toimub taaskord graafiliselt ikoonide lohistamise teel. Erinevalt teistest graafilistest



programmeerimiskeskkondadest, võimaldab LABview saada reaajas tagasisidet otse NXT-lt [10].

**leJOS NXJ** on Java rakendus NXT tarbeks. See kasutab standardset Java programmeerimiskeelt, kuid omab vähem klasse. Klasside arvu on piiratud seetõttu, et NXT 256KB mälu lihtsalt ei mahutaks sellise hulga klasside kasutust nagu Javas. Vajalik on NXT juhtploki tarkvara vahetamine leJOS tarkvara vastu. See tarkvara sisaldab Java virtuaalmasinat. NXJ programmid kirjutatakse ning kompileeritakse arvutis traditsioonilises Java keeles. Valmis kirjutatud programm saadetakse NXT juhtplokki ning seal see Java virtuaalmasina abil täidetakse [11].

**Python** on üldotstarbeline interpreteeritav programmeerimiskeel, mida algselt arendati skriptimiskeeleks. Python võimaldab mitut programmeerimistiili: objektorienteeritud, protseduurset või funktsionaalset programmeerimist [12]. Objektorienteeritud programmeerimine (OOP) on programmeerimise stiil, mis kasutab "objekte" – andmestruktuure, mis koosnevad andmeväljadest ning meetoditest [13]. Protseduurne programmeerimine (*procedural programming*) on programmeerimise stiil, mis rõhutab programmi jaotamist alamprogrammideks (protseduurideks ja funktsioonideks) [14]. Funktsionaalne programmeerimine on programmeerimise stiil, mis rõhutab funktsioonide rakendamist väärtustele [15].

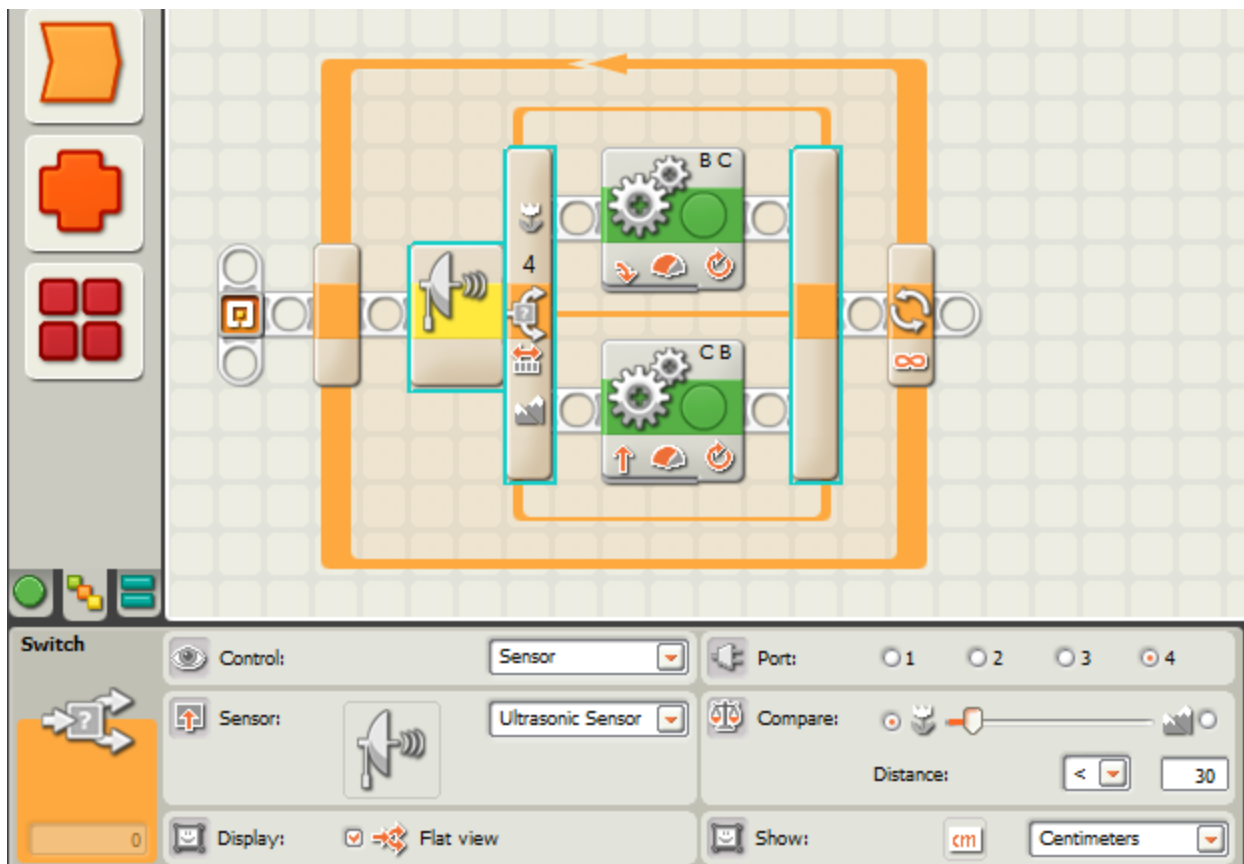
LEGO Mindstorms NXT juhtploki programmeerimiseks keeles Python on loodud NXT\_Python teek. Teegi abil on võimalik NXT juhtplokiga suhelda nii USB kui sinihamba ühenduse abil. Programmi tööpõhimõte on selline, et saadetakse käske NXT juhtplokile. Koodi ei saa kompileerida ning NXT juhtplokile laadida [16, 17].

**NBC** (*Next Byte Code*) on programmeerimiskeel, mis põhineb assemblerkeele süntaksil. Assemblerkeel on teise põlvkonna madalataseme programmeerimiskeel, mille käsud vastavad üksüheselt sihtprotsessori masinkoodi käskudele. Assembler asendab masinkäskude kahendkoodid mnemoonilise sümboliga ( tavaliselt kahe, kolme või neljatäheline inglise keelest tulev lühend ). Antud keele puhul on tegu lühikeste käskude ja nende parameetritega [18].

Esimene programmeerimiskeel, millega enamus LEGO Mindstorms robotit programmeerima hakkavad, on tavaliselt NXT-G. Nüüdseks on juba paljudesse Eesti üldhariduskoolidesse sisse viidud õppeainena robotika kursus. Sellega alustatakse tavaliselt juba põhikoolis [19]. Paljudel õpilastel on tahtmine ka robotika kursusega keskkoolis jätkata. Probleem seisneb aga selles, et selleks ajaks oskavad õpilased juba NXT-G keeles programmeerida ning sellelt tasemelt väga edasi areneda ei anna. Keerulisemaid programme on NXT-G's ebamugav kirjutada, kuna need ei mahu ära ühele ekraanitäiele ja nende täitmise kiirus ei vasta ootustele. Üle oleks vaja minna uuele tasandile, kus õpilased saaksid robotit programmeerides omandada uue keele, millest ka tulevikus neile kasu võiks olla. Järgnevas osas tulebki juttu sellest, miks on vaja üle minna teisele programmeerimiskeelele ning milline programmeerimiskeel võiks olla õpilase jaoks teiseks keeleks, mida omandada.

## **1.1. Üleminek teisele programmeerimiskeelele**

NXT-G on graafiline programmeerimisliides, mis võimaldab lihtsasti programme luua ning neid kiirelt NXT juhtimisblokki laadida. Programm moodustub üksteise järel asetatud ikoonidest, mille vahel luuakse erinevaid ühendusi [20]. Kergemaid programme on selle keskkonnaga lihtne luua (näiteks joonisel 3 on kujutatud programm, kus robot sõidab edasi ning samal ajal kontrollib sonariga eesolevaid objekte. Kui objekt on lähemal, kui 30 cm, siis toimub otsesõitmise asemel pööramine).



**Joonis 3** Programm, mis on kirjutatud NXT-G keeles. Robot sõidab sonari järgi ning kontrollib eesolevaid objekte. Kui objekt on lähemal, kui 30 cm, siis toimub otsesõitmise asemel pööramine.

Õpilase jaoks, kes on juba programmeerimisega kokku puutunud (koolis mõne aasta robotikat õppinud), võib NXT-G keel muutuda igavaks ja keerukamate programmide puhul ebamugavaks kasutada. Näiteks programmi puhul, kus robot lahendab sudokut, on roboti programmeerimiseks NXC keel võrreldes NXT-G keelega parem kasutada ( YouTube keskkonnas asuv video, kus robot lahendab sudokut ) [21]. Pikemate programmide puhul ei ole võimalik NXT-G keeles keerukamat programiosa korraga ühele ekraanitäiele kuvada. See on halb sellepärast, et nii on programmi haldamine ja sellest ülevaate saamine raske. Hea näite saab tuua Mihkel Vunki koostatud bakalaureusetööst “LEGO Mindstorms NXT’ga ühilduv käsidünamomeeter” lk 23-27, kus üks NXT-G näteprogramm on 7 ekraanitäit [22].

Ühe võimalusena, pärast NXT-G keele omandamist, ongi LEGO MINDSTROMS programmeerimiseks kasutada tekstipõhist NXC (*Not eXactly C*) keelt, millega on parem luua

keerukamaid programme ning mis ei nõua nii palju ressursse. Baitkoodi teisendatud NXC programm on võrreldes NXT-G keelest baitkoodi teisendatud sama funktsiooni täitva programmiga väiksema mälumahuga [2].

NXC on tekstipõhine programmeerimiskeel, mis on mõeldud LEGO Mindstorms NXT roboti programmeerimiseks. Keel koosneb kindlast NXC keele süntakist, mida tuleb järgida programmi kirjutamisel ning NXC API'st (*Application Programming Interface*), milles kirjeldatud funktsioonid, konstandid ja makrod, mida programmi kirjutamisel saab kasutada [23].

Järgnevas peatükis tuuakse välja NXC keele plussid.

## 1.2. NXC keel teise programmeerimiskeelena

Keeli, milles NXT juhtplokki programmeerida saab, on väga palju. Tuntumaid neist on ära kirjeldatud esimeses peatükis. ROBOLAB-i ja LABview-i puhul toimub programmeerimine samuti ikoonide lohistamise teel nagu NXT-G puhul. Nende keelte kasutamine teise programmeerimiskeelena poleks väga jätkusuutlik. NXT programmeerimiseks keeles Java on välja töötatud rakendus leJOS NXJ. leJOS NXJ püsivara tuleb aga NXT juhtplokki väikmälule paigaldada asendades LEGO Mindstorms püsivara. Halb on see sellepärast, et nii ei saa samaaegselt ühte ja sama robotit kasutada erineva tasemete õpilaste õpetamiseks. NXC põhineb assemblerkeele süntaksil, mida on raske õppida. NXT\_Python ei ole nii põhjalik, näiteks NXT\_Python teegis puuduvad moodulid LCD ekraanile info väljastamiseks ning juhtplokki nuppude jaoks. Lisaks ei saa NXT\_Pythoni abil roboti niioelda iseseisvalt tegutseda, vaid peab omama pidevat ühendust arvutiga. Erinevalt vabavaralisest NXC rakendusest on RobotC tasuline tarkvara. NXC keel on võrdlemisi lihtsasti õpitav ning võrreldes NXT-G keeles kirjutatud programmidega, on sellega parem luua keerulisemaid programme. Käesolevas peatükis tuuakse välja põhjused, miks NXC-d peaks teise keelena õppima.

Esiteks NXC keel on C programmeerimiskeelele väga sarnane. C keeleperekond on Java kõrval üks levinumaid programmeerimiskeeli maailmas. C loodi algselt Bell'i laboratoorses

õppeinstituudis. C on üks vanemaid keeli, millest vanemad mis siiani kasutuses on Fortran, Lisp ja Basic. C on avaldanud suurt mõju teiste keelte täiustumisele [24].

Kuna NXC on C keelele väga sarnane, siis järgnevalt on kirjeldatud C keele tugevad küljed.

1. C on avatud standardiga keel. C keelt saab kasutada väga paljude erinevate seadmete programmeerimiseks. Enamik teisi programmeerimiskeeli on patenteeritud ning neid saab kasutada ainult tootja poolt konkreetse riistvara programmeerimiseks (näiteks programmeerimiskeeled PIC BASIC, National Instruments Labview, assembler). Põhjus, miks riistvara müüjad pakuvad ainult oma seadmete jaoks arendatud programmeerimiskeelt on eelkõige selles, et vastava arenduskeskkonna ja programmeerimiskeele selgeksõppimine nõuab aega. Kui inimene on juba piisavalt palju aega tarkvarasüsteemi õppimisse investeerinud, siis on tal stiimul jätkata kasutamist kindla tootja riistvara, kuna ta on selgeks õppinud seadme programmeerimise antud programmeerimiskeeles.

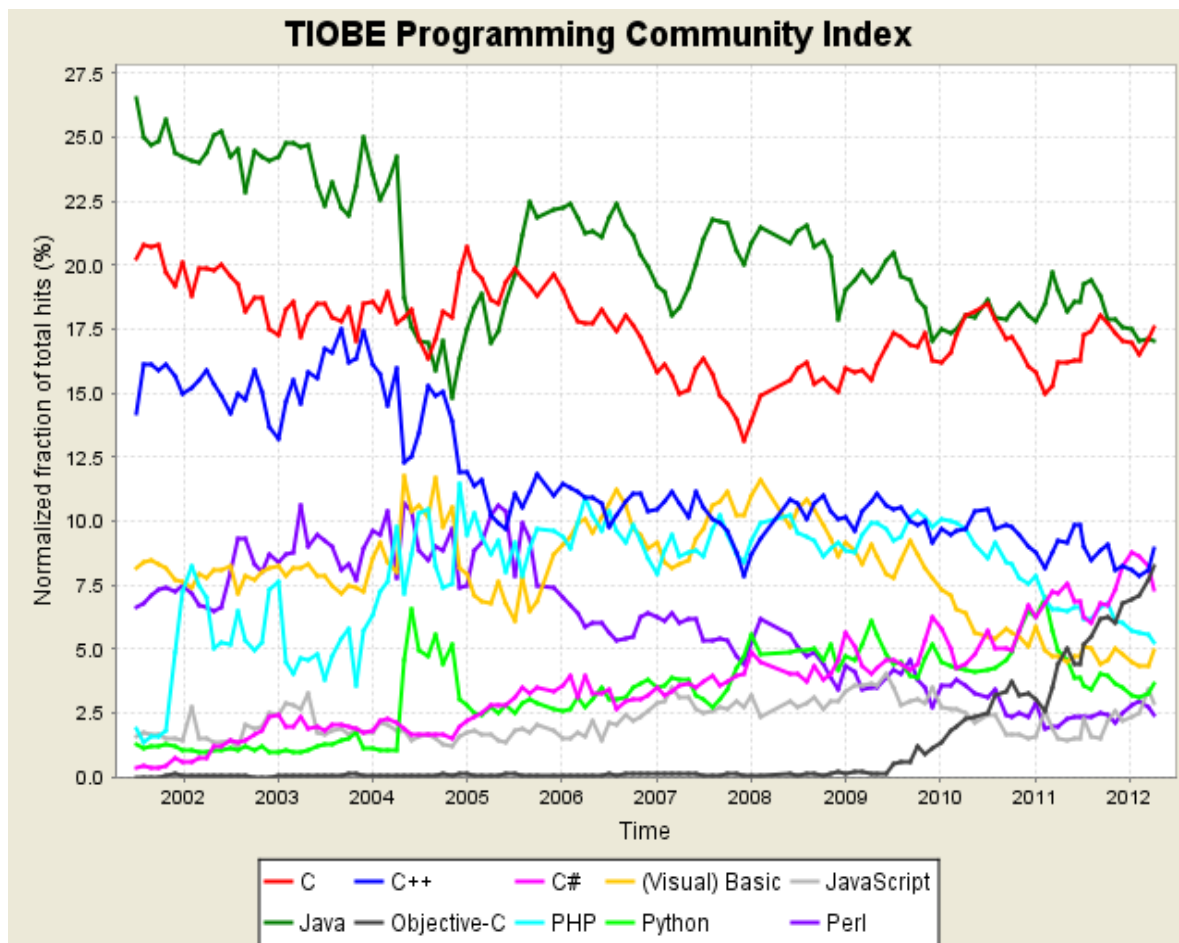
C kood on aga väga kergesti ülekantav. Suur osa koodist ei ole riistvaraspetsiifiline. See tähendab seda, et paljusid funktsioone saab kasutada erinevate riistvarade ja mikrokontrollerite peal [25].

2. C on universaalne keel. C keelt saab kasutada nii manusseadmete kui suurte rakenduste programmeerimiseks. Kui õppida selgeks programmeerimine C keeles, siis see on hea eeldus selleks, et osata kirjutada programme ka teistes programmeerimiskeeltes [25].
3. C on kiire. Ainuke võimalus kirjutada C-st kiiremat prgrammi, on kirjutada programm assembleris, mis on raskesti arusaadav ja õpitav. Erinevates keeltes kirjutatud programmide ülesannete täitmise kiirust ei tohiks alahinnata. Osade kõrgtaseme programmeerimiskeelte kiiruste erinevus on marginaalselt väike ( C++, Pascal, Fortan), aga võrreldes selliste programmeerimiskeeltega nagu Perl ja Visual Basic, siis on erinevus märkimisväärselt suur. Allolev tabel (vt tabel 1) näitab 50 000 täisarvu valikumeedodil sorteerimiseks kulunud aega. Test sooritati Mac arvuti peal, millel oli sees 2.0 GHz Core Duo protsessor ja operatsioonisüsteemiks Mac OS X Tiger [25].

Language	Run time (seconds)
C	4.01
C++	4.07
Java with JIT	6.14
Java without JIT	64.74
Perl	589

**Tabel 1** Erinevate programmeerimiskeelte kiiruse võrdlus [25]

4. C on kõige rohkem kasutatav programmeerimiskeel infotööstuses. Akadeemiliste asutuste ülesandeks on õpetada tehnoloogiaid, mida kasutatakse ka laialdaselt reaalses maailmas. Jooniselt 4 on näha, et C keel on 2012 aasta 14. novembri seisuga kõige populaarsem. Joonis on koostatud TIOBE organisatsiooni poolt, mis tegeleb programmeerimiskeelte populaarsuse uurimisega. *TIOBE programming community index* on indeks, mis näitab programmeerimiskeelte populaarsust. See on arvutatud veebiotsingute arvu põhjal, mis sisaldavad ühe otsinguparameetrina keele nime. Indeks arvestab järgnevaid otsingumootoreid: Google, Google Blogs, MSN, Yahoo!, Wikipedia ja YouTube. Indeksi kohta saab täpsemalt lugeda Tiobe kodulehelt [26, 27].



**Joonis 4** Erinevate programmeerimiskeelte populaarsus [27]

Õpilaste oskused ja teadmised peaksid vastama tööandja vajadustele. Rohkem kui 90% meie arvutis töötavatest programmidest, palju mängu ja robotite tarkvarasid on kirjutatud just C või C++ keeles [26].

5. C keel on aluskeeleks, millest on välja kasvanud enamus teisi populaarseid programmeerimiskeeli. Paljud keeled nagu C++, Java, Python, Objective-C, Perl ja PHP on arendatud C'st ning kasutavad sarnast süntaksit ja funktsioone. Kui õpilane on selgeks õppinud C keele, siis see on hea eeldus selleks, et kiiresti omandada programmeerimisoskus mõnes teises keeles [26].
6. C on võimalusterohke – C keel on kõrgtaseme keel, mis pakub assembleri keele kiirust. On väga vähe asju, mida ei saa C keeles programmeerida, näiteks 99% Unix operatsioonisüsteemist on kirjutatud C's. Samuti enamuse seadmete driverid. Teiste

kõrgtasemete keelte üldine eesmärk on vähendada esialgset keele õppimisele kuluvat aega tehes esialgsed ülesanded lihtsamaks. See on atraktiivne müügiargument vähekogenud programmeerijatele. Enamik neist aga taipab lõpuks, et nad ei saa arendada kogu tarkvarasüsteemi antud keeles erinevate võimaluste puudumise ja jõudluse piirangute tõttu [25].

Üheks plussiks, miks võiks kasutada programmeerimiseks NXC keelt, on kindlasti ka see, et NXT põhiplokil ei ole vaja vahetada tarkvara. Nii NXC kui ka NXT-G keeles kirjutatud programm kompileeritakse samasse baitkoodi ning NXT põhiplokil asuv baitkoodi interpretaator võimaldab kompileeritud programmi käivitada. Vastupidiselt programmeerimiskeele leJOS NXJ puhul, mis on Java keele sarnane, tuleb aga NXT juhtploki tarkvara vahetada leJOS tarkvara vastu [5, 28].

NXC keelt saab kasutada mitmesuguste praktiliste probleemide lahendamiseks. Programmeerimine arendab õpilase mõtlemisoskust. Kirjutades programmi, mis lahendab mõnda praktilist ülesannet või probleemi, suureneb õpilase loogiline mõtlemine ja matemaatilised oskused.

Paljud ülikoolid õpetavad teise keelena C keelt (näiteks London City University) [29]. Programmeerimine on oluline oskus sellistel erialadel nagu teadus, tehnoloogia, tehnika ja matemaatika. NXC programmeerimiskeele oskamine valmistab õpilased paremini ette ka teiste õpitavate ainete jaoks, mis on seotud kuidagi programmeerimisega (näiteks matemaatika, informaatika ja füüsika). Seega, kui õpilane, kes on NXC keelt õppinud, läheb pärast keskkooli lõpetamist ülikooli näiteks informaatikat õppima, siis on tal C keele õppimine kindlasti kergem [26].

Teiseks õpitavaks programmeerimiskeeleks NXT-G järel peaks olema NXC, kuna see annab hea eelduse C keele omandamiseks ja ülikoolis programmeerimisainetega hakkama saamiseks. Paljud tänapäeva programmeerimiskeeled on pärit kaugest sugulasest C-st. Mitmetel on osaliselt sarnane süntaks - C++, C#, Java, JavaScript, PHP, Perl. Eeliseks on ka see, et NXT põhiplokile



ei ole vaja paigalda uut tarkvara. Järgnevas peatükis kirjeldatakse täpsemalt NXC keeles programmide loomist ja käivitamist.

### 1.3. NXC keele tutvustus

NXC on C keele sarnane programmeerimiskeel, mis on kohandatud töötama LEGO Mindstorms põhiplokil. Käesolev peatükk annab ülevaate NXC programmi loomise ja käivitamise etappidest.

Robotisse ei ole vaja paigaldada uut tarkvara, vaid roboti programmeerimiseks saab kasutada nii NXC kui ka NXT-G keelt. Mõlemas keeles kirjutatud programmi kood kompileeritakse samasse baitkoodi. LEGO Mindstorms NXT programmeritavale põhiplokile on tehase poolt eelpaigaldatud baitkoodi interpretaator, mida saab kasutada programmide käivitamiseks. Sama baitkoodi interpretaatorit kasutatakse ka NXT-G graafilises keskkonnas loodud ja sealt baitkoodi kompileeritud programmide puhul [30].

NBC (*Next Byte Code*) kompilaator tõlgib NXC keeles kirjutatud programmi NXT jaoks arusaadavaks baitkoodiks, mida seejärel saab NXT põhiplokis käivitada. NXC programmi loomist ning selle käivitamist LEGO Mindstorms roboti peal iseloomustavad järgmised sammud (vt joonis 5) [36].

1. Arvutis lähtekoodi loomine ja salvestamine NXC keeles
2. Arvutis lähtekoodi kompileerimine NBC kompilaatoriga, mille tulemuseks on NXT baitkood
3. NXT baitkoodi laadimine arvutist NXT juhtploki üle USB või sinihamba ühenduse
4. NXT baitkoodi käivitamine robotis



**Joonis 5** NXC programmi loomist ning selle käivitamist LEGO Mindstorms roboti peal iseloomustavad järgmised etapid

Lähtekoodi kompileerimine NBC kompiilaatori poolt jaguneb järgmisteks alametappideks [31]:

1. Kompilaator töötleb kõigepealt makrod (eelprotssessor)
2. Kompilaator kontrollib koodi süntaksit
3. Kompilaator teisendab koodi abstraktseteks operatsioonide jadaks
4. Koodi optimeerimine
5. Objekt-failide genereerimine

NXC keele eelprotssessor tegeleb peamiselt koodi lisamisega (*#include*) ja makrodega (*#define*). Eelprotssessor ei avasta koodist endast vigu [31].

Kuigi NXC keel on väga sarnane C keelele, ei ole NXC üldotstarbeline programmeerimiskeel. Sellel on palju piiranguid, mis tulenevad NXT baitkoodi interpretaatorist [23]. Järgnevalt on toodud nõuded, millega NXC programmi loomisel tuleb arvestada.

Esiteks. NXT ühes programmis saab maksimaalselt olla 256 lõime. NXC programm koosneb erinevatest plokkidest ja muutujatest. Plokkidest eristatakse tegumeid (*tasks*) ja funktsioone (*function*). Tegumid on programmi osad, mis saavad robotis töötada paralleelsete lõimedena. Funktsioon on instruksioonide kogum, mis täidab mingit kindlat ülesannet ja seda saab vajadusel erinevates programmi osades välja kutsuda. Erinevalt tegumist saab funktsioonile ette anda parameetreid [30 lk 3-6].

Teiseks. NXT mälu on piiratud. Väikmälu on põhiplokil 256 kB. Seda kasutatakse programmide, graafika ning helifailide salvestamiseks. Kasutajale on vaba mälu umbes 120 kB. Selline mälu maht tuleneb sellest, et väikmälu paikneb ka püsivara. Tänu sellele on võimalik seadmele laadida teisi püsivarasid ja kasutada programmeerimiseks ka teisi programmeerimiskeeli. RAM mälu on 64 kB, kuhu salvestatakse programmide täitmisel informatsioon. Programmi töö lõpetamisel, see mälu tühjendatakse [32].

Kokkuvõtvalt NXC on C keele sarnane programmeerimiskeel. NXC keele kompilaator tõlgib NXC keeles kirjutatud programmi NXT jaoks arusaadavaks baitkoodiks, mida seejärel on võimalik NXT juhtploki peal käivitada.

## 1.4. NXC keele süntaksist

NXC keel on keele loomisest peale mõeldud asenduseks assemblerile. Lisaks on pandud väga suurt rõhku koodi loetavusele. Eelmises peatükis anti ülevaade NXC programmi loomise ja käivitamise etappidest. Antud peatükk keskendub NXC keele süntaksi tutvustamisele.

Muutuja tüübi muutmine väiksema maksimumväärtusega tüübilt suuremale ei kujuta reeglina ohtu. Näiteks *unsigned int* tüüpi muutuja (16 bitine) muutmisel *unsigned long* tüüpi muutujaks (32 bitine) ei kujuta piisava mälu juures ohtu. Muutuja tüübi muutmisel suurema maksimumväärtusega tüübilt väiksema maksimumväärtusega tüübile peab alati tegema eelneva kontrolli [31].

NXC keeles kasutatavad muutujate tüübid on välja toodud tabelis 2 [30 lk 6-7].

Tüübi nimetus	tüüp	Andmed
<b>bool</b>	Loogika	0 või 1 (false või true)
<b>byte, unsigned char</b>	Täisarv	0 kuni 255
<b>char</b>	Tähemärk	Unicode-sümbolid
<b>unsigned int</b>	Täisarv	Väärtus saab olla vahemikus 0 kuni 65 565
<b>short, int</b>	Täisarv	Väärtus saab olla vahemikus -32 768 kuni 32 767
<b>unsigned long</b>	Täisarv	Väärtus saab olla 0 kuni 4 294 967 296
<b>long</b>	täisarv	Väärtus saab olla -2 147 483 648 kuni 2 147 483 647
<b>float</b>	Ujukomaarv	Näiteks jagamistehte tulemus 0,6667
<b>mutex</b>	Mutex	Spetsiaalne muutujatüüp paralleeltöötamise jaoks
<b>string</b>	Sõne	Sõne ehk tekst
<b>array</b>	Erinevad	Jada

**Tabel 2** Muutujate tüübid NXC keeles

Muutujate kasutamiseks tuleb need eelnevalt programmis defineerida või ka lihtsamini öeldes valmis teha. Muutujad defineeritakse vastava muutujatüübiga. Muutujaid saab jaotada lokaalseteks ja globaalseteks. Lokaalsed deklareeritakse koodiploki sees. Need on kasutatavad alates deklareerimise kohast kuni konkreetse ploki lõppemiseni. Globaalsed muutujad luuakse väljaspool koodiplokke. Globaalseid muutujad ei ole mõtet algväärtustada nulliks. Programmi käivitudes nullitakse eelnevalt kõik globaalsed muutujad ära ning peale nullimist kopeeritakse vastavatele aadressidele algväärtused [33].

Tabelis 3 on välja toodud NXC keeles enimkasutatavad aritmeetilised operaatorid ja täiendatud omistustehted [30 lk 15]. Täiendatud omistamine (*augmented assignment*) näitab muutust sama muutuja väärtuses, kus omistusmärgile on lisatud tehtemärk. Näiteks:  $n = n + 1$  tähendab, et hetkel kehtivat  $n$ -i väärtust suurendatakse 1 võrra ja tehte tulemus säilitatakse samas muutujas.  $n += 1$  on täiendatud omistamine ja tähendab seda sama.

Tehe	Näide	Seletus
<b>Omistamine</b>	$x = y;$	Omistab muutujale väärtuse.
<b>Liitmine</b>	$c = a + b;$ $a++;$ $a += 10;$	$a++;$ on sama mis $a = a+1;$ $a += 10;$ on sama mis $a = a+10;$
<b>Lahutamine</b>	$c = a - b;$ $c --;$ $c -= 10;$	$c--;$ on sama mis $c = c-1;$ $c -= 10;$ on sama mis $c = c-10;$
<b>Korrutamine</b>	$c = a * b;$	Näiteks $x *= y;$ See on samaväärne tehtega $x = x * y;$
<b>Jagamine</b>	$c = a / b;$	Näiteks $x /= y;$ See on samaväärne tehtega $x = x / y;$
<b>Jäägi leidmine</b>	$c = a \% b;$	Näiteks $x \% = y;$ See on samaväärne tehtega $x = x \% y;$
<b>Aboluutväärtus</b>	$a = \text{abs}(b);$	$  =$ omistab muutujale absoluutväärtuse omistatavast Näiteks $x   = y;$
<b>Kas on võrdne?</b>	$a == b$	Tehte tulemus on kas tõene või väär, ehk <i>True</i> või <i>False</i>
<b>Kas ei ole võrdne?</b>	$a != b$	Tehte tulemus on kas tõene või väär, ehk <i>True</i> või <i>False</i>
<b>Väiksem võrdne</b>	$a <= b$	Tehte tulemus on kas tõene või väär, ehk <i>True</i> või <i>False</i>
<b>Suurem võrdne</b>	$a >= b$	Tehte tulemus on kas tõene või väär, ehk <i>True</i> või <i>False</i>
<b>Suurem kui</b>	$a > b$	Tehte tulemus on kas tõene või väär, ehk <i>True</i> või <i>False</i>
<b>Väiksem kui</b>	$a < b$	Tehte tulemus on kas tõene või väär, ehk <i>True</i> või <i>False</i>
<b>Loogiline eitus</b>	$a = !b$	Kui $b$ on väär siis tehte tulemusena on $a$ väärtus tõene Näiteks $x = \text{true};$ ja $y = !x;$ , siis $y = \text{false};$
<b>Loogiline "JA"</b>	$a \&\& b$	Kui $a$ on tõene ja $b$ on tõene on tehte tulemus tõene. Teiste juhtudel on tehte tulemus väär
<b>Loogiline "VÕI"</b>	$a    b$	Kui üks kahest, kas $a$ või $b$ on tõene, siis on ka tehte tulemus tõene. Kui mõlemad, $a$ ja $b$ on väär, on ka tehte tulemus väär.

**Tabel 3** NXC keeles kasutatavad aritmeetika ja loogika tehteid

Täiendatud omistustehete puhul ei tohi erinevate tehtemärkide vahele tühikut lisada. Kui kasutada näiteks tehet  $x += y;$ , siis see ei ole korrektne ja kompilaator annab veateate.

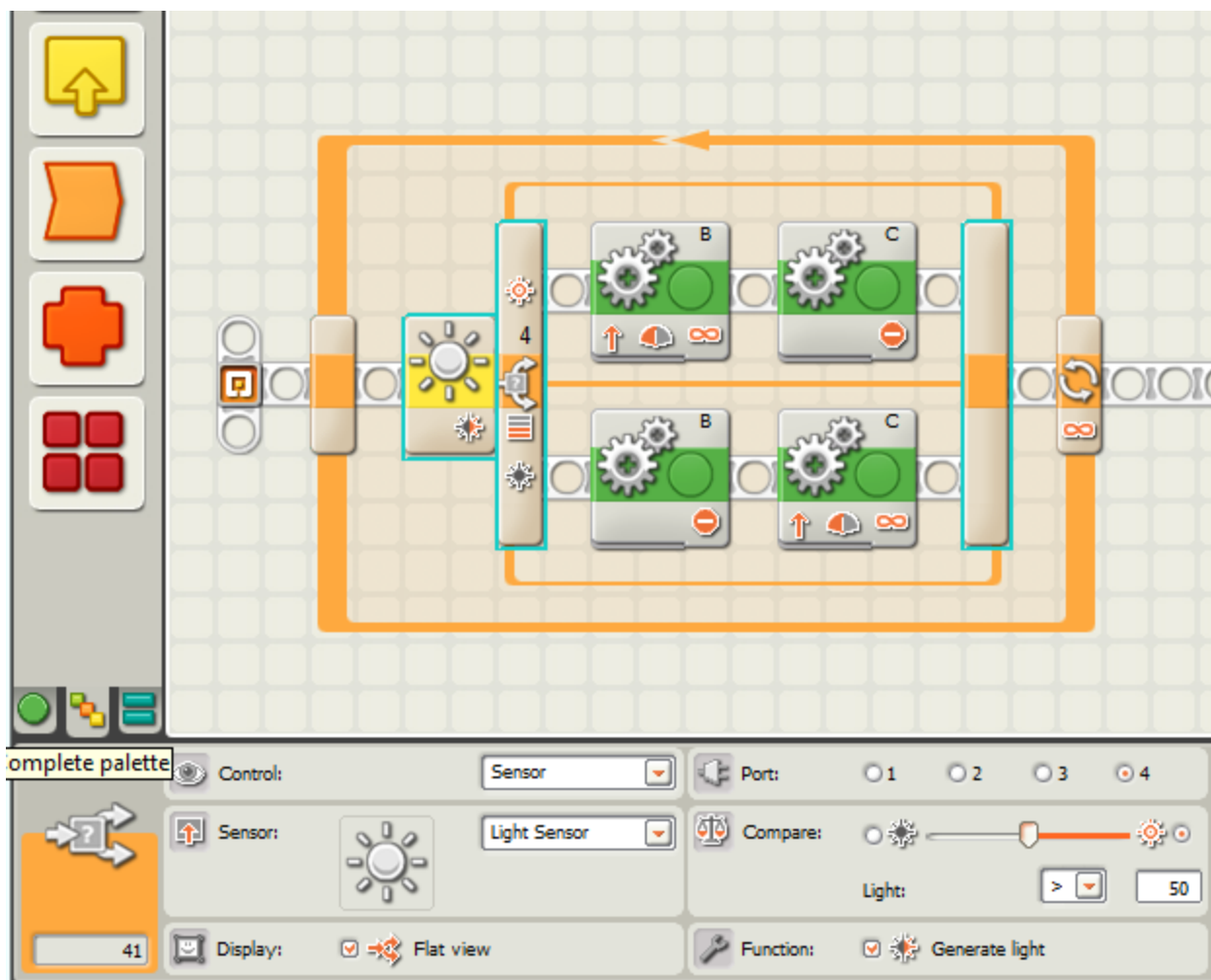
Programmeerimiskeeles NXC on kõik funktsioonid võrdväärsed. Erinevalt programmeerimiskeelest PASCAL ei saa siin defineerida mingi funktsiooni sees omakorda uusi funktsioone, mis oleksid ka kättesaadavad ainult sellele funktsioonile. Kordloodud funktsioonid on NXC keeles kasutatavad kõigi teiste funktsioonide poolt. Ainult funktsiooni *main()* ei tohi enam teiste funktsioonide poolt välja kutsuda [34].

Käesolev peatükk andis põgusa ülevaate NXC keeles kasutatavatest muutujatest ja loogika ning aritmeetika tehetest. Täpsemalt NXC keele leksikaalsest ehitusest, programmi struktuurist, NXC operaatoritest ja juhtstruktuuridest ning funktsioonidest saab lugeda antud magistritöö raames loodud Moodle kursuselt “LEGO Mindstorms NXT roboti programmeerimine keeles NXC”. Õppematerjalist on tehtud ka .pdf väljavõtte, mis asub käesoleva töö lisades CD plaadil (vt. Lisa 2).

## **1.5. NXT-G ja NXC keelest baitkoodi teisendatud programmide mälukasutuse võrdlus**

Punktis 1.3 oli juba põgusalt juttu sellest, et NXT põhiploki mälu on piiratud. Väikmälu on põhiplokil 256 KB. Seda kasutatakse programmide, graafika ning helifailide salvestamiseks. Väheste põhiploki mälu tõttu peab arvestama, et programmid, mida laetakse põhiplokki, ei läheks mahult liiga suureks. Järgnevalt on võrreldud baitkoodi teisendatud NXC ja NXT-G sama funktsioone täitvate programmide suurust.

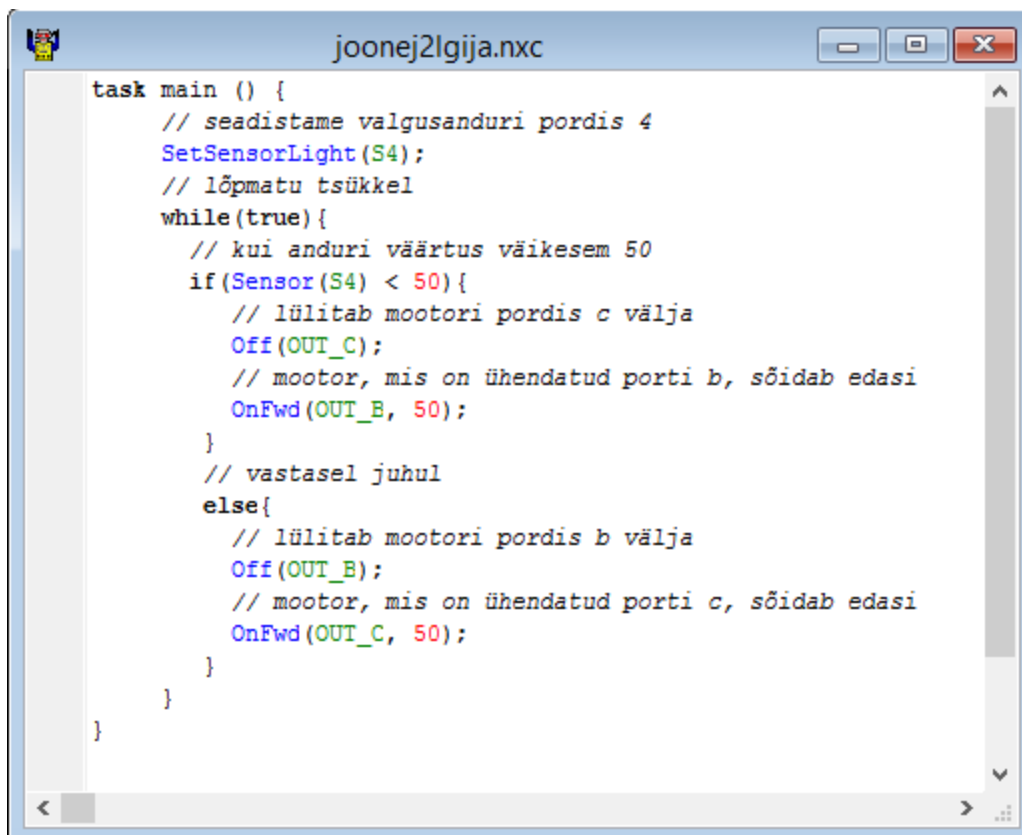
Jooniselt 6 on näha NXT-G keeles kirjutatud programm, kus robot sõidab mööda joont.



**Joonis 6** NXT-G keeles kirjutatud joonel sõitva roboti programm

Kogu programmi sisu asub tsüklis, mille sees kontrollitakse valgusanduri väärtust, mis on ühendatud porti number neli. Kui anduri väärtus on suurem 50-st, siis liigutatakse mootorit, mis on ühendatud porti B ning peatatakse mootor pordis C. Kui väiksem, siis liigutatakse mootorit pordis C ning peatatakse mootor pordis B.

Jooniselt 7 on näha NXC keeles kirjutatud programm, mille ülesanne on sama, mis joonisel 6 kujutatud NXT-G keeles programmi puhul.



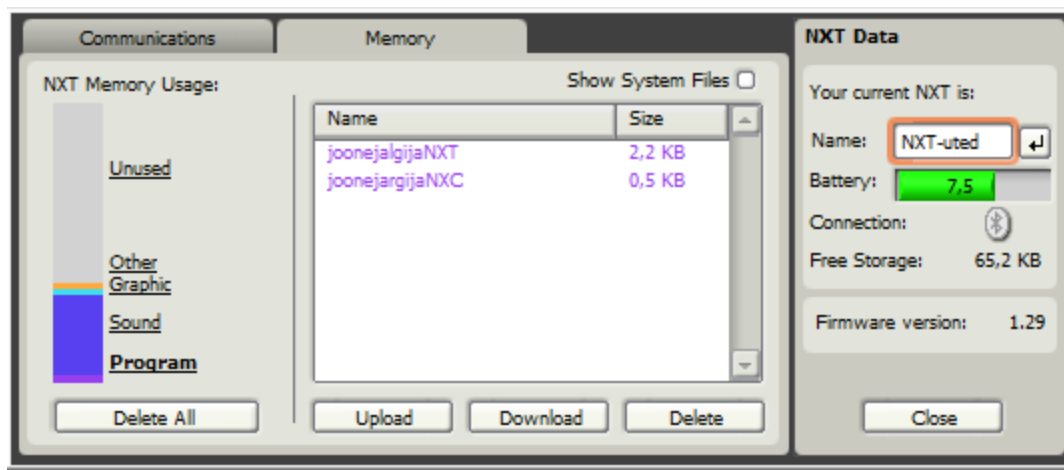
```
task main () {
    // seadistame valgusanduri pordis 4
    SetSensorLight(S4);
    // lõpmatu tsükkel
    while(true){
        // kui anduri väärtus väikesem 50
        if(Sensor(S4) < 50){
            // lülitab mootori pordis c välja
            Off(OUT_C);
            // mootor, mis on ühendatud porti b, sõidab edasi
            OnFwd(OUT_B, 50);
        }
        // vastasel juhul
        else{
            // lülitab mootori pordis b välja
            Off(OUT_B);
            // mootor, mis on ühendatud porti c, sõidab edasi
            OnFwd(OUT_C, 50);
        }
    }
}
```

**Joonis 7** NXC keeles kirjutatud joonel sõitva roboti programm

*SetSensorLight(S4)*-ga määratakse, et porti 4 on ühendatud valgusandur ning *Sensor(S4)* loetakse valgusanduri väärtust pordis 4. Vastavalt valgusanduri väärtusele lülitatakse vastav mootor *Off* funktsiooniga välja ja *OnFwd* funktsiooniga pannakse edasi liikuma.

Jooniselt 8 on näha, et baitkoodi teisendatud NXC programm on võrreldes NXT-G keelest baitkoodi teisendatud sama funktsiooni täitva programmiga oluliselt väiksema mälumahuga.





**Joonis 8** NXT-G ja NXC keelest baitkoodi teisendatud programmide mälukasutuse suurused

NXT-G keelst baitkoodi kompileeritud programmi suurus on 2,2 KB, NXC keelest baitkoodi kompileeritud programmi suurus on 0,5 KB, mis on ligikaudu 4 korda väiksem, kui NXT-G-st kompileeritud koodi suurus. See võimaldab juhtploki korraga laadida suuremaid programme.

Antud peatükk võrdles kahte sama funktsiooni täitvat programmi, millest üks oli kirjutatud NXT-G ja teine NXC keeles.

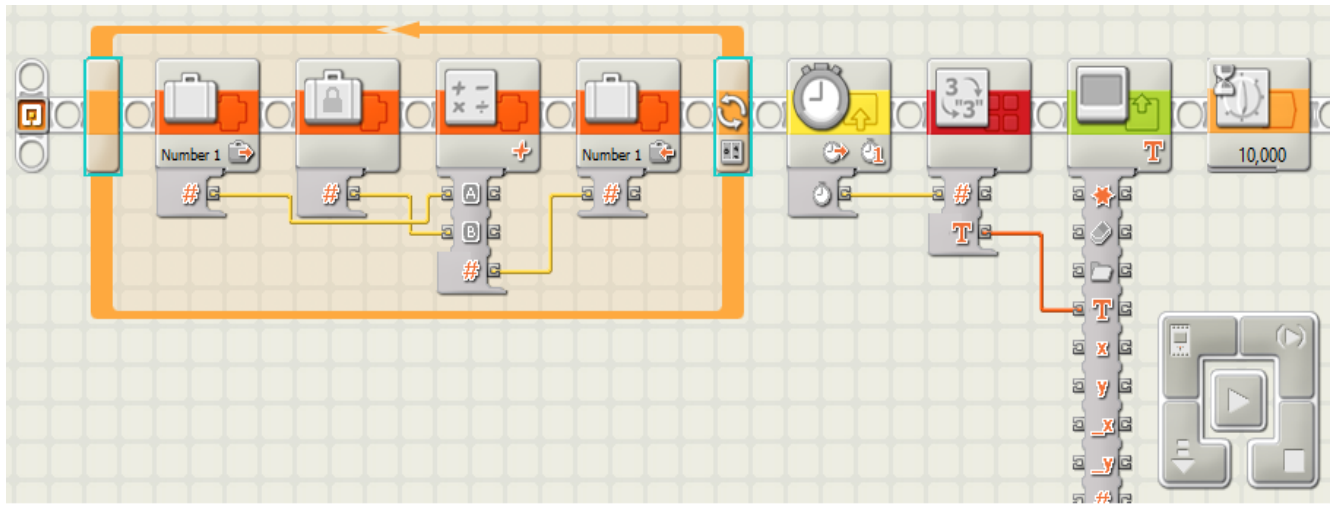
## 1.6. NXT-G ja NXC keeles kirjutatud programmide täitmise kiiruse võrdlus

Alapeatükis 1.2, kus loetleti põhjuseid, miks peaks teiseks programmeerimiskeeleks olema NXC, toodi ühe plussina välja see, et NXC keeles kirjutatud programm on kiirem. Selles peatükis tuleb täpsemalt võrdluse alla NXT-G ja NXC keeles kirjutatud sama funktsiooni täitvate programmide kiiruse võrdlus.

On programme, mille korral programmi täitmise kiirus mängib olulist rolli. Näite võib tuua 3D printimise valdkonnast, mille korral tuleb detaili ettevalmistamisel teha suur hulk erinevaid arvutusi ning seejärel hakatakse õhukeste kihtide kaupa 3D kujutist tekitama. YouTube keskkonnast on leitav ka video pealkirjaga “LEGO 3D Milling Machine - 3D Printer”, mille korral on kasutatud NXT juhtploki 3D kujutise väljalõikamiseks [35].

Võrdlusena oleks hea olnud kasutada massiivide sorteerimist näiteks valikmeetodil, aga kuna NXT-G keelel puudub ametlik tugi massiivide kasutamiseks, siis otsustati võrdlusmeetodina kasutada arvude liitmist. Liitmiseks kasutati tsükli, mille korduste arvu muudeti. Tsükli sees liideti summale iga kord +1 juurde. Lõpus väljastati arvude liitmiseks kulunud aeg millisekundites ekraanile.

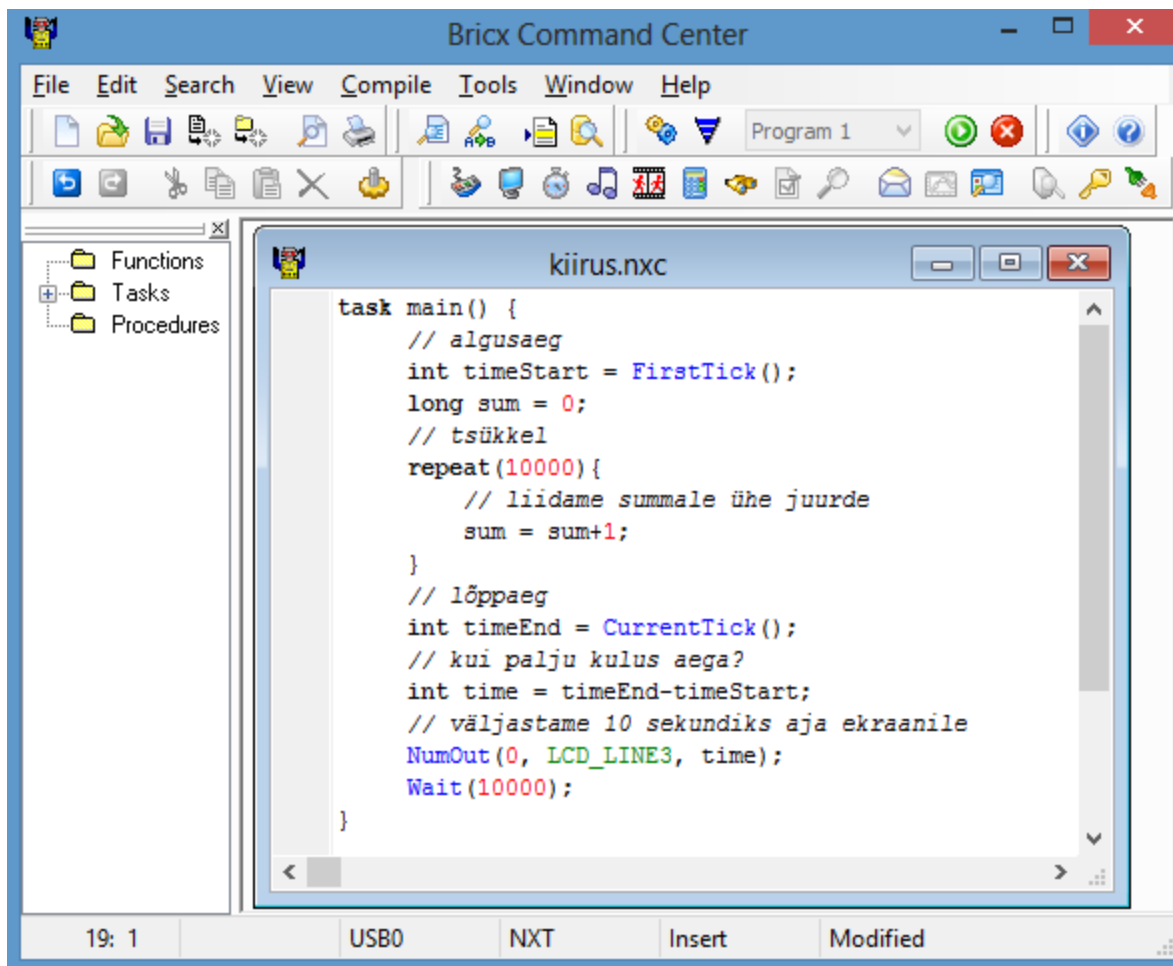
Joonisel 9 on kujutatud NXT-G keeles kirjutatud programm, mis mõõdab aega, mis kulub arvude liitmiseks tsükliks.



**Joonis 9** NXT-G keeles kirjutatud programm, mis mõõdab aega, mis kulub arvude liitmiseks tsükliks ning väljastab seejärel kulunud aja millisekundites ekraanile

Programmi esimene osa koosneb tsüklist, mille sees liidetakse muutujale iga kord juurde konstant, mille väärtus on 1. Seejärel väljastatakse tsükli läbimiseks kulunud aeg NXT juhtploki ekraanile kümneks sekundiks. NXT baitkoodi teisendatud programmi suuruseks on 1,1 KB.

Joonisel 10 on kujutatud NXC keeles kirjutatud sama funktsiooni täitev programm.



**Joonis 10** NXC keeles kirjutatud programm, mis mõõdab aega, mis kulub arvude liitmiseks tsüklis ja väljastab kulunud aja millisekundites ekraanile

Muutujale *timeStart* omistatakse väärtus algusaeg ning muutujale *timeEnd* omistatakse hetkeaeg pärast tsükli *repeat* läbimist. Kulunud aeg tsükli läbimiseks leitakse lahutades hetkeajast algusaeg. Tulemus väljastatakse funktsiooniga *NumOut* roboti ekraanile kümneks sekundiks. Antud programmi suurus pärast baitkoodi teisendamist on 0,3 KB.

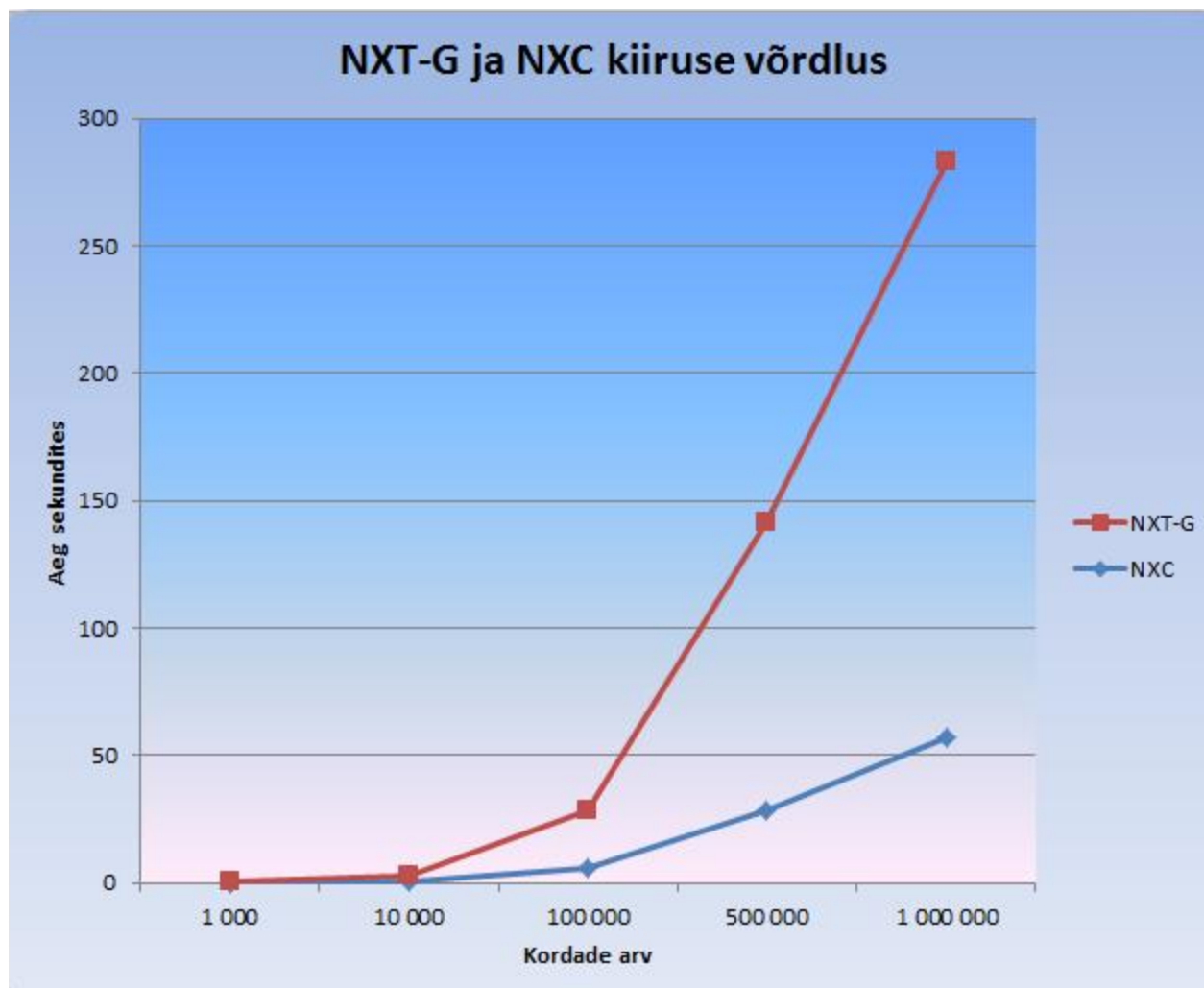
Tabelis 4 on välja toodud ajad sekundites, mis kulusid erineva arvu tsüklite läbimiseks NXT-G ja NXC keeles kirjutatud programmidel.

Nr	Tsükli arvu	Kulunud aeg NXT-G sekundites	Kulunud aeg NXC sekundites
1.	1 000	0,227	0,060
2.	10 000	2,269	0,571
3.	100 000	22,602	5,681
4.	500 000	113,057	28,394
5.	1 000 000	226,110	56,785

**Tabel 4** NXT-G ja NXC keeles kirjutatud programmide arvude liitmiseks kulunud aegade võrdlus

Tsükli sees toimuvaks tegevuseks oli igal etapil ühele täisarvule +1 juurdeliitmine kasvatades niiviisi arvu igal etapil ühe võrra.

Tulemustest parema ülevaate saamiseks on loodud ka graafik (vt joonis 11).



**Joonis 11** Graagikult on võrreldud NXT-G ja NXC keeles kirjutatud programmide aegu, mis kulusid liitmistehete sooritamiseks.

Graafikult on selgelt näha, kuidas tsükli kordade arvu suurendamisel kasvab NXT-G keeles kirjutatud programmi täitmisaeg võrreldes NXC keeles kirjutatud programmi täitmisaajaga märgatavalt. Vahe on igal etapil umbes neljakordne. Programmi puhul, kus kiirus mängib olulist rolli, on see üheks väga oluliseks argumendiks, miks peaks programmi kirjutamiseks kasutama NXC keelt.

Käesolevad peatükis võrreldi NXT-G ja NXC keeles kirjutatud sama funktsiooni täitvate programmide kiiruseid. Tulemuseks saadi, et NXC keeles kirjutatud programmi täitmiskiirus on märgatavamalt kiirem.

## 2. NXCEesti

NXCEesti on Priit Ranna poolt loodud eestikeelne keskkond NXC kasutamiseks, mis töötab nii Windows, Linux kui ka Mac OS operatsioonisüsteemidel [36 lk 29]. Tarkvara võimaldab kasutada põhilisi tekstiredaktori võimalusi ning omab igale platvormile sobivat kompilaatorit redigeeritava programmi kompileerimiseks ja robotisse saatmiseks. Igal platvormil on oma kompilaator, mille õige kasutamise eest vastutab programm.

Programm kasutab oma tööks andmeid, mis on salvestatud tekstifailidesse. Järgnevalt tuleb juttu antud programmi paigaldamisjuhistest.

### 2.1. NXCEesti installeerimisjuhend

NXCEesti kasutamiseks esitatavad minimaalsed nõuded arvutile on [36 lk 31-32]:

1. 27MB vaba kõvakettaruumi
2. 70MB vaba operatiivmälu
3. USB 2.0 port
4. Operatsioonisüsteem Windows, Linux või Mac OS
5. JRE 6 ( Java Runtime Environment 6 ) [37]
6. Windows ja Mac OS operatsioonisüsteemide puhul on nõutav Fantom Driver [38]

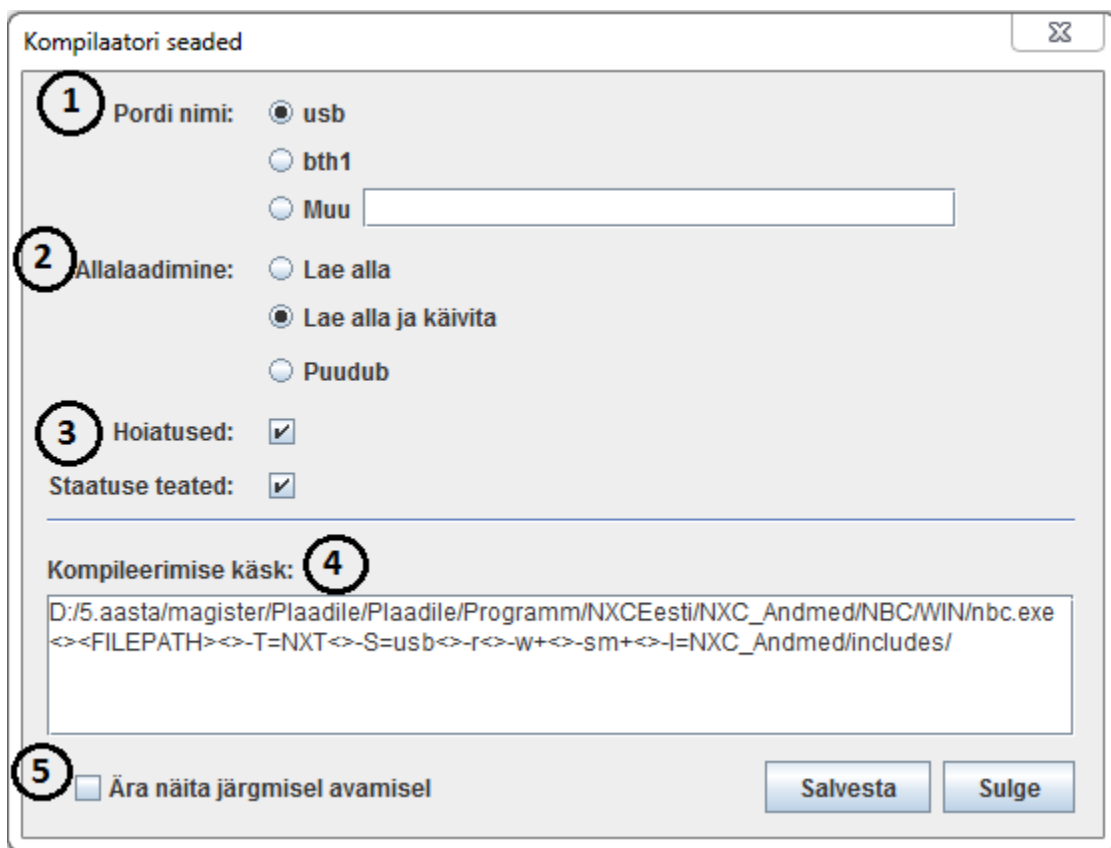
Programmi paigaldamiseks tuleb alla laadida antud Moodle kursuselt .zip fail, mille leiab kursuse pealehelt nimega NXCEesti Programm. Kui fail on alla laetud, tuleb see lahti pakkida ning avada kaust NXCEesti. Rakenduse tööle panemiseks tuleb käivitada fail NXCEesti.jar

Esmasel käivitamisel kuvatakse kompilaatori seadete aken (vt joonis 12).

1. Kõigepealt saab valida, missugust ühendusviisi roboti ja arvuti vahel kasutatakse. USB ühenduse kasutamiseks tuleb valida USB. Sinihamba (ing. k. *Bluetooth*) ühenduse loomine antud versiooni puhul robotiga ei tööta.
2. Allalaadimise valiku juures saab teha valikuid, missuguseid toiminguid tehakse pärast NXC keeles kirjutatud programmi kompileerimist. “Lae alla” valiku puhul laetakse

kompileeritud programm robotisse. Programmi käivitamiseks peab kasutaja seda robotis ise tegema. “Lae alla ja käivita” valiku puhul laetakse kompileeritud programm robotisse ja käivitatakse. “Puudub” valiku puhul programm kompileeritakse, aga robotisse seda ei saadeta. Selle valiku puhul on hea programmi testida, kas see kompileerub.

3. “Hoiatused” ja “Staatuse teated” tuleks täiendava informatsiooni saamiseks sisse lülitada, kuna need võimaldavad programmist paremini vigu otsida või eduka kompileerimise puhul tagasisidet saada.
4. Automaatselt on kuvatud ka kompileerimise käsk, mille tee peab viitama NBC kompilaatorile. Antud näites kasutati Windows operatsioonisüsteemi ning seetõttu viidatakse ka Windows operatsioonisüsteemi jaoks mõeldud kompilaatorile.
5. “Ära näita järgmisel avamisel” valikuga määratakse, kas järgmisel programmi käivitamisel näidatakse kompilaatori seadete akent või mitte.



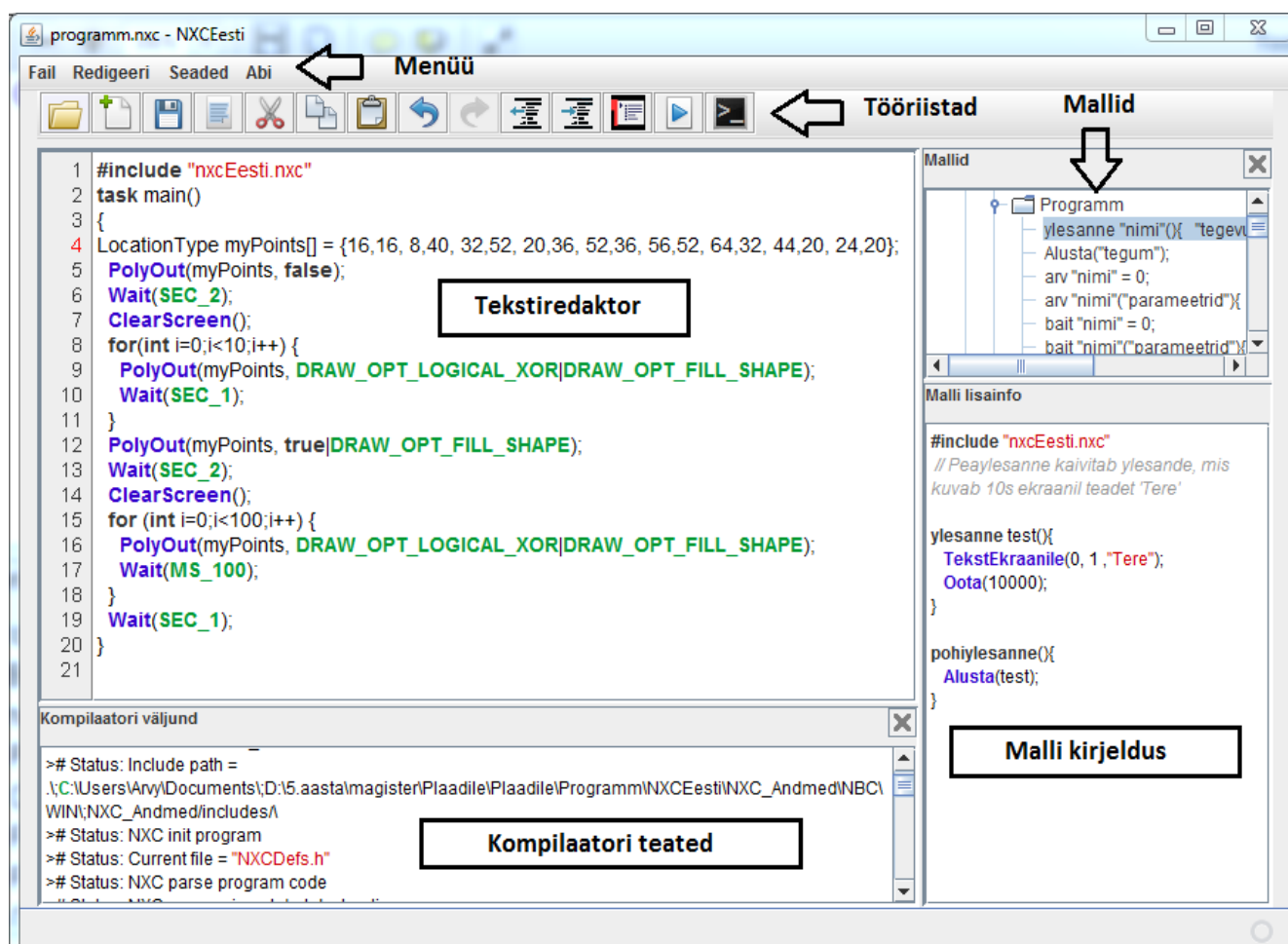
**Joonis 12** NXCEesti programmi seadete aken

Kompilaatori seadeid on võimalik ka hiljem muuta. Selleks tuleb valida rakenduse seadete menüüst “Kompilaator” valik.

Käesolev peatükk keskendus NXCEesti paigaldamisjuhendile ning kompilaatori seadetele. Järgmises peatükis tuleb juttu rakenduse kasutajaliidest.

## 2.2. NXCEesti kasutajaliides

Programmi NXCEesti kasutajaliidese vaade on kujutatud joonisel 13. Kasutajaliides jaguneb eri osadeks: menüü, tööriistad, mallid, mallide kirjeldused, kompilaatori teated. Järgnevalt kirjeldatakse neid osi täpsemalt [36 lk 33-38].



Joonis 13 NXCEesti kasutajaliides



- **Peamenüü** valikuteks on: Fail, Redigeeri, Seaded ja Abi. Need menüüd sisaldavad alamvalikuid, mis on vajalikud programmi seadete muutmiseks, failide salvestamiseks ja muutmiseks ning abiinfo saamiseks.
- **Tööriistariba** sisaldab lisaks menüüdes olevatele valikutele ka võimalusi programmis kasutatavate paneelide näitamiseks ja NXC keeles kirjutatud koodi kompileerimiseks.
- **Mallid** on NXCEesti programmi üks osa, mis sisaldab endas koodi kirjutamise struktuurinäiteid, kuidas midagi väärtustada või milliseid parameetreid mingi funktsioon nõuab.
- **Malli kirjelduse** all kuvatakse lisainfo konkreetse valitud malli kohta, kui see on olemas. Lisaks on osade mallide kasutamise kohta toodud koodinäited.
- **Tekstiredaktor** on ala, kus toimub uue koodi kirjutamine.
- **Kompilaatori teadete** all kuvatakse programmi kompileerimise käigus tekkinud vea- ja staatusteadete.

Peatükk tutvustas NXCEesti peamisi kasutajaliidese osi. Järgmises peatükis tuleb lähemalt vaatluse alla rakendusliidese eestikeelsed tõlked.

## 2.3 NXCEesti eestikeelsete funktsioonide teek

NXCEesti kasutab oma tööks andmeid, mis on salvestatud tekstifailidesse. Failid asuvad kaustas **NXC\_Andmed** ja on kättesaadavad ka läbi redaktori. Valides menüüst **Seaded** → **Redaktori falid** → **Märksõnad** avatakse fail, mis sisaldab endas märksõnu, mis värvitakse paksus (*bold*) kirjas. Eestikeelsetest märksõnadest näiteks: **korda** - *repeat*, **kuni** - *while*, **kui** - *if*, **pohiylesanne** - *task main*. Valides menüüst **Seaded** → **Redaktori falid** → **Konstandid** avatakse fail, mis sisaldab endas konstante, mis värvitakse programmis vaikimisi rohelist värvi. Näiteks on konstantide alla lisatud väljundportide tähised: **A**, **B**, **C**, **AB**, **AC**, **BC**, **ABC**. Fail **NXC API** sisaldab endas NXC keele funktsioone, mis värvitakse programmis vaikimisi sinist värvi. Eestikeelsetest funktsioonidest näiteks: **Heli**("port");, **KeeraParemaleP**("kraadi", "kiirus");, **PeataMootor**("port");. Soovi korral saab märksõnade, konstantide, funktsioonide ja sõnade värvi programmis muuta. Selleks tuleb valida menüüst **Seaded** → **Tekst**. Failidesse saab ka

värvitavaid märksõnu ise juurde lisada, erinevad sõnad tuleb üksteisest eraldada reavahetusega [36 lk 39-40].

Programmile saab kasutaja lisada uusi tõlkeid ja standardfunktsioone. Selleks tuleb täiendada faili **NXCEesti.nxc**. Programmist on see kättesaadav valides menüüst **Seaded** → **Redaktori failid** → **NXCEesti API**. Tegemist on NXC programmiga, millel puudub tegum *main*. Failis on tõlked loodud kasutades makrode defineerimist ja erinevat tüüpi funktsioone. Selleks, et antud tõlkeid ja standardfunktsioone kasutada, tuleb iga uue loodava NXC programmi päisesse lisada rida *#include "nxcEesti.nxc"*. See laeb programmi faili **nxcEesti.nxc**, milles on defineeritud tõlked [36 lk 40].

Makrode defineerimine käib käsuga *#define*. See võimaldab programmfaili päises luua identifikaatoriga tekstiosasid. Enne programmi kompileerimist asendatakse vastav identifikaator defineeritud tekstiosaga. Makrosid luuakse struktuuriga *#define identifikaator defineeritav*, kus identifikaatoriks on nimi, mida kasutatakse makro asukoha määramiseks programmis ja defineeritavaks on NXC programmiosa, mis enne kompileerimist makro nime asemele pannakse [36 lk 13, 30]. Järgnevalt on toodud näide makro kasutamisest.

*//Ootab etteantud aja*

*#define Oota(x) Wait(x) // Makro*

*task main () {*

*int aeg = Oota(10000); // Enne preprotsessorit*

*int aeg = Wait(10000); // Pärast preprotsessorit*

*}*

**Näide 1** *Makro kasutamise näide*

NBC (*Next Byte Code*) sisaldab endas preprotsessorit, mis töötleb programmi lähteteksti faile, enne kui neid reaalselt kompileerima hakatakse [39].

Samas failis, **NXCEesti.nxc**, on defineeritud ka eestikeelsed funktsioonid, mida programmis kasutada saab. Näited tõlgitud funktsioonidest:

```
// Mootor B edasi etteantud kraadide ja kiirusega
inline void Paremp(int kraadi = 360, int kiirus = 50){
    RotateMotor(OUT_B, kiirus, kraadi);
}
```

**Näide 2** NXCEesti tõlgitud funktsiooni näide, mis liigutab mootorit B etteantud kiiruse ja kraadi võrra

Antud funktsiooni puhul on vaikimisi väärtustatud ka **kraadi** ja **kiirus** parameetrid. Seega kui programmis kutsuda funktsioon välja ilma parameetriteta ( `Paremp();` ), siis kasutatakse vaikimisi väärtusi.

```
// Vaigistab heli
inline void VaigistaHeli(){
    StopSound();
}
```

**Näide 3** NXCEesti tõlgitud funktsiooni näide, mis vaigistab heli.

**Inline** märksõna kasutamisel tehakse kompilaatoris funktsiooni kasutamisel sellest koopia. See tähendab seda, et iga kord kui vastavat funktsiooni välja kutsutakse, luuakse funktsiooni koodist koopia vastavasse kohta, kus seda funktsiooni välja kutsuti. Selle puuduseks on see, et koopiategemisel suureneb baitkoodi maht. Plussiks on aga see, et funktsiooni väljakutsumisele kulub vähem aega. Täpsema ülevaate kõigist tõlgitud märksõnadest ja funktsioonidest saab loodud Moodle kursusel “LEGO Mindstorms NXT roboti programmeerimine keeles NXC” ning töö lisana kaasasolevast CD plaadilt õppematerjali peatüki 3.9 alt (vt. Lisa 2).

Mallide paneelil asetsevaid malle saab muuta menüü valikuga **Seaded** → **Mallid**. Valides valikutest mingi malli, kuvatakse selle sisu aknas **Mall** ning kirjeldus aknas **Kirjeldus**. Mallid on jaotatud kahte peakausta: **nxcEesti** ja **NXC**. NxcEesti kaustas asuvad mallid, milles on kirjeldatud kõik eestikeelsed tõlked ja funktsioonid. Parema ülevaate saamiseks on need omakorda jaotatud alamkaustadesse. Näiteks: **Mootorid**, kus asuvad kõik mootoritega seotud toimingud ja **Ekraan**, mis hõlmab endas ekraaniga seotud toiminguid. Teises kaustas, NXC, on

kirjeldatud NXC API funktsioonid, muutujad ja tingimuslaused ning tsükliid. Mallide muutmiseks tuleb valida vastav mall, teha tekstiredaktoris vastavad muudatused ning vajutada nuppu **Muuda**. Mallide lisamiseks tuleb valida mingi astme element ning vajutades nuppu **Lisa**, lisatakse valitud elemendi järele uus sama astme element. Malle on võimalik ka kustutada. Mallide andmed salvestatakse **NXC\_mallid.txt** faili [36 lk 38].

Käesolev peatükk andis ülevaate NXCEesti tekstifailidest, mida programm oma tööks kasutab ning eestikeelsete funktsioonide loomise põhimõttest. Järgmises peatükis näidatakse, kuidas loodud tõlkeid kasutada.

## 2.4 NXCEesti näidisprogramm

Järgnevalt on toodud kaks näiteprogrammi, millest esimene on loodud kasutades NXC ingliskeelseid API funktsioone (näide 4) ja teine kasutades NXCEesti tõlkeid (näide 5).

```
/*
Andurite näide
*/
task main()
{
    //Seadistame valgusanduri NXT kolmandas sisendis
    SetSensorLight(IN_3);
    while(true)
    {
        //Kas valgusanduri väärtus on väiksem kui 50?
        if(Sensor(IN_3)<50)
        {
            // tagurdab
            OnRev(OUT_BC, 50);
            Wait(1500);
            // Lülitab mootori pordis B välja
            Off(OUT_B);
            Wait(500);
        }
        else
        {
            // Sõidab otse edasi
            OnFwdSync(OUT_BC, 100, 0);
        }
    }
}
```

```

    }//while lõpp
} //main lõpp

```

#### Näide 4 NXC andurite programm

```

/*
Andurite näide kasutades NXCEesti tõlkeid
*/
pohiylesanne()
{
    //Seadistame valgusanduri NXT kolmandas sisendis
    ValgusSisseValgusega(3);
    kuni(toene)
    {
        //Kas valgusanduri väärtus on väiksem kui 50?
        kui(Valgus(3)<50)
        {
            // Tagurdab
            OtseS(BC, 0, -50);
            Oota(1500);
            // Lülitab mootori pordis B välja
            PeataParem();
            Oota(500);
        }
        muidu
        {
            OtseS(BC, 0, 100);
        }
    }
} //while lõpp
} //main lõpp

```

#### Näide 5 NXC andurite programm kasutades NXCEesti tõlkeid

Meeles tuleb pidada seda, et NXCEesti eestikeelsete funktsioonide kasutamisel määratakse sisendpordid lihtsalt numbriga ja väljundpordid vastava tähega. Näiteks:

- **HeliSisse**(4); // Seadistatakse sisendport number 4 kuulama heliandurit
- **OtseS**(BC, 0, 100); // Pannakse roboti mootorid, mis on ühendatud porti B ja C edasi sõtma

Kiiruse parameetrid saavad olla vahemikus -100 kuni 100. Kui kiirus on negatiivse märgiga, siis liigub mootor vastassuunas. Sensorite ja mootorite portide ning ekraani rea väärtused ei saa olla muutujates. Väärtused tuleb otse funktsiooni kirjutada.

Selles peatükis tutvustati NXCEesti rakendusliideses kirjutatud NXC prorgammi kasutades NXCEesti tõlkeid. Järgnev osa kirjeldab antud magistritöö raames loodud Moodle kursuse loomise protsessi.

### 3. NXC e-kursuse loomise protsess

Järjest enam töötatakse Eesti üldharidus-, kutse- ja ülikoolides välja e-õppematerjale ja e-kursusi eesmärgiga rikastada traditsioonilist õpet. E-kursuse loomisega on seotud erinevad sihtrühmad: õppija, õpetaja, arendaja ja administratsioon. Õpetajate peamine soov on teha huvitav kursus, mis samas vastaks ka õppija vajadustele ning õppeasutuse ootustele. Õpetajate aeg on aga piiratud ning enamus nende ajast kulub õppetundide andmiseks.

Vajadus uue kursuse järgi on olemas, kuna NXT-G keele võimalused on piiratud ning seda on keerukamate programmide puhul ebamugav kasutada. Üldhariduskooli õpetajal pole aga töö kõrvalt uue kursuse loomiseks lihtsalt aega. Kuna valdkond on uus, siis peaks õpetajad eelnevalt selle teema endale selgeks tegema. Praegu ei leidu eesti keeles NXC keele kohta ühtegi e-kursust ning teadmisi antud keele kohta tuleks otsida väga erinevatest kohtadest või lihtsalt API-t (*Application Programming Interface*) lugedes ning need on võõrkeelse tekstiga. API on rakendusliides, mis kujutab endas reeglistikku olemasoleva valmisprogrammiga suhtlemiseks.

Antud magistritöö eesmärgiks ongi luua e-kursus eelkõige õpetajakoolitusel osalejatele, kes saaksid kursuse raames mõistliku aja jooksul vajalikud teadmised NXC keele kohta omanda ning omandatud teadmisi ja oskusi seejärel õpilastele edasi anda. Koolitusel osalejate alla kuuluvad õpetajad, kes osalevad RAJU keskuse poolt korraldavatel täiendkoolitustel ja vajavad õppematerjalidele ligipääsu nii täiendkoolituse ajal kui ka pärast selle lõppu.

Hea ülevaate e-kursuse loomise etappidest ja selle vastamisest kvaliteedinõuetele annab e-Õppe Arenduskeskuse poolt loodud käsiraamat „Juhend kvaliteetse e-kursuse loomiseks“ [40]. Käsiraamatu eesmärgiks on pakkuda tuge e-kursuste kavandamisel ja loomisel ning olemasolevate e-kursuste hindamisel. Raamat on ainulaadne, kuna selles käsitletakse terviklikult kogu kursuse elutsükli etappe: analüüs, õppeprotsessi kavandamine, kursuse väljatöötamine, kursuse läbiviimine ja kursusele hinnangu andmine. Raamat jagunebki peatükkideks vastavalt eelkirjeldatud etappidele. Iga peatüki lõpus on nimetatud olulised kvaliteedinõuded, mis peavad olema täidetud, et tunnistada e-kursus ja selle õpetamise protsess parimatele nõuetele vastavaks.

## **Kursuse korralduslikud aspektid**

Uue kursuse loomisel on otsustava tähtsusega kursuse põhjalik planeerimine. Samuti tuleb lähtuda sihtrühma võimekusest ning keskkonna tehnoloogilistest võimalustest. NXC e-kursuse eesmärgiks on anda põhilised teadmised NXC keele struktuurist ja ehitusest ning roboti programmeerimisest keeles NXC. Kursus jaguneb neljaks osaks:

I osa – sissejuhatus

II osa – Juhendid ja abimaterjalid

III osa – NXC keele tutvustus (keele süntaks)

IV osa – Ülesanded (ülesanded ja näited) Antud osa eesmärk on pakkuda valikut ülesandeid, mida kursusel osaleja saab lahendada kas hinnatavate ülesannete raames või lihtsalt harjutamise jaoks.

Järgnevas peatükis kirjeldatakse kursuse eesmärke ja nõudeid kursusele.

## **3.1 E-kursuse analüüs**

Analüüs on kogu kursuse loomise protsessi algus. Selle käigus lahutatakse terviklik teema lihtsamateks osadeks, et luua seoseid ning lihtsustada teemast arusaamist. Toimub vajaduste, sihtrühma ja konteksti analüüs.

Analüüsi etapi saab jagada neljaks sammuks [40]:

- 1. vajaduste analüüs ja üldiste eesmärkide sõnastamine**
- 2. tingimuste analüüs**
- 3. õppijate ehk sihtrühma analüüs**
- 4. sisu analüüs**

Järgnevalt on iga sammu analüüsitud loodava kursuse kontekstist lähtudes.

### **1. Vajaduste analüüs ja üldiste eesmärkide sõnastamine**



Vajadus uue kursuse järgi on olemas, kuna roboti programmeerimise võimalused NXT-G graafilises keskkonnas on piiratud ning õpilaste jaoks, kes on juba programmeerimisega kokku puutunud, võib NXT-G muutuda igavaks kasutada.

**Vajalikeks eelteadmisteks on:**

1. Baasteadmised LEGO Mindstorms NXT robotist
2. Teadmised roboti baaskomplekti osadest
3. Teadmised roboti tarkvarast ja programmeerimisest keeles NXT-G

Nõutud eelteadmiste omandamiseks sobib kõige paremini LEGO Mindstorms NXT algkursus. Kursus asub samuti Moodle keskkonnas ning selle kursuse raames on võimalik omandada kiiresti baasteadmised LEGO Mindstorms NXT robotikakomplektist.

Kuna õpetatav teema on uus, siis puuduvad ka enamustel robotika õpetajatel teadmised NXC keelest. Seega on vajalik antud teemat kõigepealt õpetajatele selgitada. Antud e-kursus ongi mõeldud eelkõige õpetajakoolitusel osalejatele, kes saaksid vajalikud teadmised NXC keelest ning seejärel kasutades antud kursust tugimaterjalina, annaks omandatud teadmised ja oskused edasi õpilastele.

Kursus peab tähelepanu pöörama järgmistele teemadele:

1. Ülevaade NXC keelest
2. NXCEesti kasutusjuhend
3. Juhend, kuidas robotit NXC keeles programmeerida
4. Näite- ja programmeerimisülesanded
5. Võimaldada õppijal analüüsida kursuse sisu ja nõudeid, hinnata neid ja anda kursusele tagasisidet

Lisaks peab e-kursus võimaldama õppijatel õppimiseks endale valida sobiv aeg, koht ja tempo.

## **2. Tingimuste analüüs**

Kursus peaks valmima magistritöö raames ning ajaliselt on see piiratud magistritöö valmimise ajalise tähtajaga, milleks on 2013 aasta maikuu keskpaik. Täiendusi kursuse sisu muutmiseks on

võimalik kursuse läbiviijatel teha ka hiljem. Peamiseks kursuse loojaks on antud magistritöö autor. Nõu ja ettepanekutega on abiks antud magistritööd juhendajad Anne Villemis ja Taavi Duvin, kes omavad pikaldasi ja häid kogemusi robotikast ja õpetamisest.

Kursust hakkavad läbi viima MTÜ Robotika koolitajad, kes koolitavad välja robotikaõpetajaid erinevates üldhariduskoolides. Kursusele on täienduseks auditoorne õpe, kus toimub kursusest ülevaate andmine ja selle sisu põgus tutvustus. Samuti on võimalik näidata ette lahendatud ülesandeid. Veebipõhine keskkond võimaldab iseseisvalt õppida ning vajadusel pärastpoole ligipääsu kursuse materjalidele.

Tehnoloogilistest vahenditest on olemas LEGO Mindstorms NXT robotid, mida on võimalik programmeerida. Õpperühma suurusel otsest piirangut pole, kuna tegu on e-kursusega. Materjalid tuleb ise koostada ja saab kasutada teiste lõputööde tekste. Tuleb koostada ka test kontrollimaks baastadmisi, tagasiside küsimustik ning kursuse haldamise juhend. Kursus on mõeldud intensiivkursusena 3-4 nädalase perioodi peale. Kasutatav skeem oleks selline, et päevasele või kahepäevasele intensiivkursusele, kus kõik viibivad füüsiliselt koos, järgneb määratud pikkusega kaugkoolitusperiood ning kursuse lõpuks kogunetakse jälle kokku lõppseminariks.

### **3. Õppijate ehk sihtrühma analüüs**

Õppijateks on õpetajakoolitusel osalevad õpetajad, kellel on juba õpetamist lubav dokument käes ning kes on eelnevalt õpetamisega kokku puutunud. Samuti peaks neil olemas olema baastadmised LEGO Mindstorms NXT robotist. Samas peab arvestama sellega, et neil ei pruugi olla eelnevalt suurt kokkupuudet programmeerimisega tekstipõhises keskkonnas. Seetõttu peavad materjalid olema kergesti arusaadavad ka neile, kes pole programmeerimisega eelnevalt kokku puutunud.

Kuna tegemist on õpetajatega ja koolitatavad õpetajad hakkavad pärast kursuse läbimist ise seda materjali õpetama, siis on kursus otseselt seotud nende tööga. Arvestama peab ka sellega, et kursusel osalevad õpetajad peavad eeldavasti kaugkoolitusperioodi ajal ka oma tööülesannetega tegelema.

#### 4. Sisu analüüs

Loodav kursus peab olema sobiva mahuga ning sisaldama põhimaterjale antud teema kohta, mis aitavad õppijal õpiväljundeid saavutada ning vastama kursuse õpiväljunditele. Materjalidele, mis ei kuulu põhimaterjalide hulka, kuid mis võivad huvi pakkuda kursuse läbijale, kes soovib teemaga süvitsi minna, tuleks viidata viidete kogu all. Link viidete kogule saab olema kursuse pealehel. Sisu peab olema asjakohane ning arusaadavalt esitatud. Teoreetilise osa vahepeale tuleb luua praktilisi ülesandeid, kus õppija saab kohe midagi järgi proovida ja kaasa mõelda. See muudab teksti lugemise huvitavamaks ja paremini mõistetavaks.

Sisu järjestamiseks on kõige mõistlikum kasutada struktuurilisest loogikast lähtuvat järjestust, mis tähendab, et sisu on järjestatud vastavalt aine loogilisele struktuurile. Sissejuhatavas osas võrreldakse NXT-G ja NXC keelt ning tuuakse välja NXC keele eelised. Seejärel esitatakse juhendid NXCEesti [36] ja Bircx Command Center [5] programmeerimiskeskondadest. Kolmandas osas tutvustatakse NXC keele süntaksit ning viimane osa keskendub programmeerimisülesannetele.

Järgnevalt on loetletud õpiväljundid, mida kursusel osaleja pärast kursuse läbimist oskama peaks.

Pärast kursuse läbimist oskab:

1. nimetada NXC keele eeliseid NXT-G keele ees.
2. kasutada NXCEesti programmeerimiskeskonda.
3. kasutada Bircx Command Center programmeerimiskeskonda.
4. nimetada NXC programmi loomise ja käivitamise etappe.
5. nimetada NXC keele leksikaalseid reegleid.
6. kasutada NXC keele peamisi muutujaid ( *int*, *bool*, *long*, *float* ja *string* ).
7. kasutada jadasid.
8. kasutada NXC keele operaatoreid ( omistamine, liitmine, lahutamine, korrutamine, jagamine, jäägi leidmine ning erinevad loogika operaatorid ).
9. nimetada ja kasutada peamisi NXC programmi juhtstruktuure ( *while*, *for* ja *repeat* tsüklid, *if-else* tingimuslause ).

10. kasutada tuntumaid NXC keele standardfunktsioone.
11. leida NXC rakendusliidestest (*API*-st) erinevaid funktsioone ja nende seletusi.
12. kirjutada NXC keeles lihtsamaid programme.
13. kasutada NXCEesti tõlkeid.
14. NXC programmi kompileerida ning laadida seda NXT robotisse.

Nõutavaks eelteadmisteks on baastadmised LEGO Mindstorms NXT robotist ja selle juhtplokist.

### **Kursuse nõuded**

1. Peab võimaldama osaleda kursusel erinevatel huvigruppidel
  - 1.1 Õpetajakoolituse osalejad
  - 1.2 Õpilased ja üliõpilased
  - 1.3 Lõputööde kirjutajad
2. Kursuse vastamine kvaliteedinõuetele
3. Kursuse lihtne hallatavus

Antud peatükk keskendus vajaduste, sihtrühma ja kursuse sisu analüüsile. Järgnevas peatükis keskendutatakse kursuse korralduslikele aspektidele.

## **3.2 Õppeprotsessi kavandamine**

Käesolevas peatükis pööratakse tähelepanu teemadele, mida peab arvesse võtma kursuse loomise etapis.

Kursus koosneb ühest baasmoodulist, mis jaguneb nejaks osaks: sissejuhatus, NXC keele programmeerimiskeskondade juhendid ja abimaterjalid, NXC keele tutvustus ning praktilised ülesanded. Baasmaterjali peale tuleb sooritada üks test omandatud teadmiste kontrollimiseks. Lisaks tuleb lahendada kaks programmeerimisülesannet (üks kergema raskusastmega, teine edasijõudnutele). Alternatiivina võib teise ülesande asemel mõelda ise välja mõni huvitav ülesanne ja see realiseerida. Mõlemad lahendused tuleb esitada õppejõule.

## Hindamiskriteeriumid

Hindamise meetodiks on mitmeeristav hindamine. Arvestuse saamiseks tuleb kokku saada vähemalt 70 punkti 100-st. Moodle test annab maksimaalselt 50 punkti. See sisaldab endas küsimusi baasmaterjali kohta. Testi tulemus peab olema vähemalt 60%, testi hinnatakse automaatselt. Programmeerimisülesanded annavad kokku 50 punkti. Teise ülesande asemel võib ka esitada enda koostatud ülesande koos lahendusega. Mõlema ülesande eest on võimalik maksimaalselt saada 25 punkti. Lisaks on võimalik saada boonuspunkte õppejõu poolt, kui õppija leiab õppematerjalidest mingi vea ja põhjendab selle ära.

## Kursuse ajakava

Tabelis 5 on nädalate kaupa esitatud kursuse ajakava.

Kuupäev	Liik	Sisu
1. nädal	Loeng	1. Kursuse tutvustus ja lühiülevaade 2. Korralduslikud aspektid
1. nädal	E-õpe	1. Lugeda läbi baasmooduli sissejuhatav I osa 2. Baasmooduli II osa – Olemasolevad vahendid NXC kasutamiseks
2. nädal	E-õpe	Baasmooduli III osa – NXC keel (peatükid 3.1 – 3.5) 1. NXC programmi loomise ja käivitamise etapid 2. NXC keele leksikaalne ehitus 3. NXC programmi struktuur 4. NXC operaatorid 5. NXC juhtstruktuurid
3. nädal	E-õpe	Baasmooduli III osa – NXC keel (peatükid 3.6 - 3.10) 1. NXC preprotsessori käsud 2. NXC rakendusliides 3. NXC valmisfunktsioonide kirjeldused 4. NXCEesti eestikeelsete funktsioonide teek 5. Näidisprogrammid <b>Sooritada Test baasteadmiste peale</b>
4. nädal	E-õpe	Baasmooduli IV osa - Ülesanded 1. Ülesanded algajatele 2. Ülesanded edasijõudnutele <b>Esitada hindamiseks 2 ülesannet</b>

<b>4. nädal</b>	<b>Seminar</b>	1. Arutlemine ning kokkuvõtete tegemine 2. Tagasiside küsimustiku täitmine
-----------------	----------------	---

**Tabel 5** *Kursuse ajakava*

Tegemist on neljanädalase intensiivkursusega. Kursuse alguses saavad kõik kokku ning tutvustatakse kursust ning selle korraldust. Sellele järgneb e-õppe periood, kus omandatakse vajalikud teadmised NXC keeles roboti programmeerimiseks ning sooritatakse nõutud ülesanded. Kursuse lõpus saadakse jälle füüsiliselt kokku ning toimub arutelu, kogemuste vahetamine, kursuse läbimise tunnistuse saamine ning kokkuvõtte tegemine.

### **Õpistiilid**

Iga inimene õpib talle ainuomasel viisil, mis on seotud tema maailmatunnetuse, tundemaailma, väärtuste ja veendumustega. Seetõttu peab e-kursuse koostaja arvestama materjalide loomisel, et kavandatavad õpitegevused ei põhineks ainult ühel õpistiilil. Peamised õpistiilid jagunevad: visuaal-verbaalne, visuaal-mitteverbaalne, audio-verbaalne, taktil-kinesteetiline, kehalis-kinesteetiline, visuaal-ruumiline, verbaal-lingvistiline ja loogilis-matemaatiline. Täpsemalt, mida mingi õpistiil endast kujutab, on lahti seletatud “Juhend kvaliteetse e-kursuse loomise” käsiraamatus [40]. Selleks, et arvestada võimalikult paljude õpistiilidega, tuleks lisaks tekstilisele materjalile kasutada näitlikustavaid graafikuid ja jooniseid ning illustreerivaid pilte.

### **Sünkroonsed ja asünkroonsed tegevused**

Üks olulisi otsuseid õppematerjali disainimisel on, kas õppurid saavad määrata millal nad õpivad või mitte, kas kasutada sünkroonseid või asünkroonseid tegevusi. Sünkroonne tegevus õppeprotsessis tähendab, et kõik tegevuses osalejad peavad oma panuse andma samaaegselt, asünkroonne tegevuse puhul võivad ajad erinevad olla.

Sünkroonseid tegevusi tuleks eelistada kui [41]:

- õppurid peavad teistega õpitavat jooksvalt arutama
- õppurid vajavad pidevat motiveerimist
- enamusel õppuritest on samad vajadused, samad küsimused

Asünkroonseid tegevusi tuleks eelistada kui [41]:

- õppurid on füüsiliselt erinevates ajatsoonides
- õppurite töögraafik ei võimalda paindlikkust
- loodud materjal on mõeldud iseõppimiseks
- õppuritel on erinevad individuaalsed vajadused

Antud kursuse raames eelistatakse rohkem asünkroonseid tegevusi, kuna e-kursus on mõeldud õpetajakoolitusele osalevatele robotika õpetajatele, kelle töögraafik ei võimalda suurt paindlikkust. Seetõttu peab õppematerjale välja töötades arvestama, et need sobiksid õppijale iseseisvaks õppimiseks ja toetamaks auditoorset kokkusaamist. E-õppe puhul saab õppija ise valida endale õppimiseks sobiva aja ja koha. See võimaldab paidlikku õppeviisi. Erinevate arutelude jaoks saab hästi kasutada kursuse foorumit. Arutelud on head abivahendid saadud info läbitöötlemiseks ja sellest arusaamise kontrollimiseks. Oluline on, et loodav õppematerjal oleks hästi mõistetav, õppeprotsessi toetav, korrektselt disainitud ning soovitatavalt vajadusele lihtsasti muudetav.

### **Õpiobjektid**

Kuna e-õppe materjalide väljatöötamine on suhteliselt ressursinõudlik, siis tuleks materjalide loomisel järgida põhimõtet, et need oleks korduvkasutatavad, mida saaks vajadusel üha uute e-kursuste materjalidele lisada. See tähendab seda, et loodav õppematerjal peab koosnema õpiobjektidest. Õpiobjekt on vähim iseseisev materjali osa, mille suurus võib varieeruda, kuid parimaks peetakse, kui üks õpiobjekt vastab ühele õpieesmärgile. Iga õpiobjekt peaks olema terviklik ning kontekstist võimalikult sõltumatu. Sõltumatus teeb kursuse erinevatest tükkidest kokkupanemise lihtsamaks ning võimaldab ühte õpiobjekti kasutada mitmetes õppetükkides või kursustes [42].

### **Õpihaldussüsteem**

Kursuse loomiseks kasutatakse õpihaldussüsteemi Moodle, mis sisaldab endas vahendeid struktureerimiseks, kursuse sisu esitamiseks (failid, õpimoodulid, meediakogu), suhtlemiseks (foorum, jututuba), õpitegevuste sooritamiseks (ülesanded, testid, rühmatöövahendid) ja kursuse haldamiseks (õpipäevik, osalemise jälgimine). Valik osutuks Moodle kasuks, kuna Tartu

Ülikooli enamik kursusi asuvad selles keskkonnas ning see pakub häid võimalusi e-kursuse loomiseks.

Kursuse loomist Moodle keskkonda ning selle seadistamist on põhjalikult käsitletud Moodle juhendis “Kuidas alustada uue e-kursuse loomisega Moodle’is, seega käesolevas magistritöös uue kursuse lisamist ning seadistamist olulisel määral ei käsitleta [43].

### 3.3 Kursuse loomine

Kursuse loomine on protsess, mille jooksul viiakse väljatöötatud sisu (õppematerjalid, testid, juhendid jm) üle e-õppesse. Sisu esitamine peab vastama nõuetele, mis on kooskõlas akadeemiliste nõuetega. E-kursusel teiste materjalide kasutamisel peab kindlasti arvestama autoriõiguse seadusega [40].

Kuna antud teema on uudne, siis tuleb kursuse loojal endal õppematerjalid kokku panna. Materjalide koostamisel on abiks Nikolai Konovalovi poolt loodud bakalaureusetöö, kust on plaanis sisse võtta eestikeelsed funktsioonide seletused ja näited [44]. Samuti on plaanis koostada juhend Priit Ranna poolt magistritöö raames loodud programmi NXCEesti kasutamise võimalustest [36].

E-kursuse õppematerjalide loomiseks kasutatakse Moodle raamatu moodulit. See on Moodle keskkonna lisamoodul, mis tuleb eraldi alla laadida ning paigalda. Kui Moodle.org lehelt on allatõmmatud soovitud moodul [45], siis tuleb see lahti pakkida ning lahtipakitud kaust tuleb paigutada *server/moodle/mod/* kataloogi. Täpsemalt on kirjeldatud uue mooduli lisamise protsessi “Moodle installeerimise ja administreerimise” õpetuse all [46]. Selleks, et lisamoodulit paigaldada, peavad kasutajal olema vastavad õigused. Tartu Ülikooli Moodle keskkonnas on antud lisamoodul juba paigaldatud.

Raamat võimaldab Moodles luua vasakul paikneva sisukorraga õppematerjali. Raamatusse saab lisada nii teksti, pilte, veebilinke kui ka audio- ja videofaile. Raamatu kohal on nupud edasi-



tagasi liikumiseks, terve raamatu või valitud peatüki väljatrükiks ning raamatu salvestamiseks pdf-failina [47].

Raamatu peatükkide lisamisel saab määrata, kas need on nummerdatud, täpiloendina või taandega. Kuna antud kursus koosneb neljast põhipeatükist, mis omakorda jagunevad alapeatükkideks, siis on kasutatud peatükkide lisamisel taandega varianti ning iga peatüki pealkirja algusesse on lisatud ka peatüki number nende paremaks eristamiseks (vaata joonis 14). Uue peatüki loomise puhul on võimalik valida, kas see on pea- või alapeatükk. Raamatu mooduli puuduseks on see, et see võimaldab tekitada vaid kuni kahetasemelisi peatükke.

Sisukord
<b>I osa - Sissejuhatus</b>
1.1 Taust
1.2 NXC ja NXT-G võrdlus
1.3 NXC keele eelised NXT-G ees
1.4 NXC arenduskeskkonnad
1.5 Näide NXC programmist
1.6 Seletus

**Joonis 14** Moodle kursuse peatükkide ülesehitus

Materjalide loomine toimub otse Moodle keskkonnas. Uue peatüki loomisel kuvatakse tekstitoimetuste aken, kus on olemas üldtuntud nupud teksti kujundamiseks, piltide, tabelite ja veebilinkide lisamiseks (vaata joonis 15).



**Joonis 15** Tekstitoimetuste aken Moodle keskkonnas

Lisaks veel nupud:

1. Wordist teksti kopeerimisel kujunduse eemaldamiseks
2. Võimalus lisada automaatset linkimist
3. Audio salvestamine otse Moodles
4. HTML koodis toimetamiseks
5. Tekstitoimetusakna suurendamiseks

Materjalid salvestatakse HTML formaadis ning seda HTML koodi on võimalik ka tekstitoimetuste akna kaudu muuta. Materjalide vormindus peab vastama RAJU keskuse poolt sätestatud stiiljuhiste [48]. Käsitsi tekstide loomine tekstiredaktoris ja seal kujundamine ning piltide lisamine tähendab suurt käsitööd, kuid kuna valmisdokumente pole, mida kohe Moodlesse vastava peatüki alla importida saaks, siis tuleb peatükid ja nende sisu käsitsi luua.

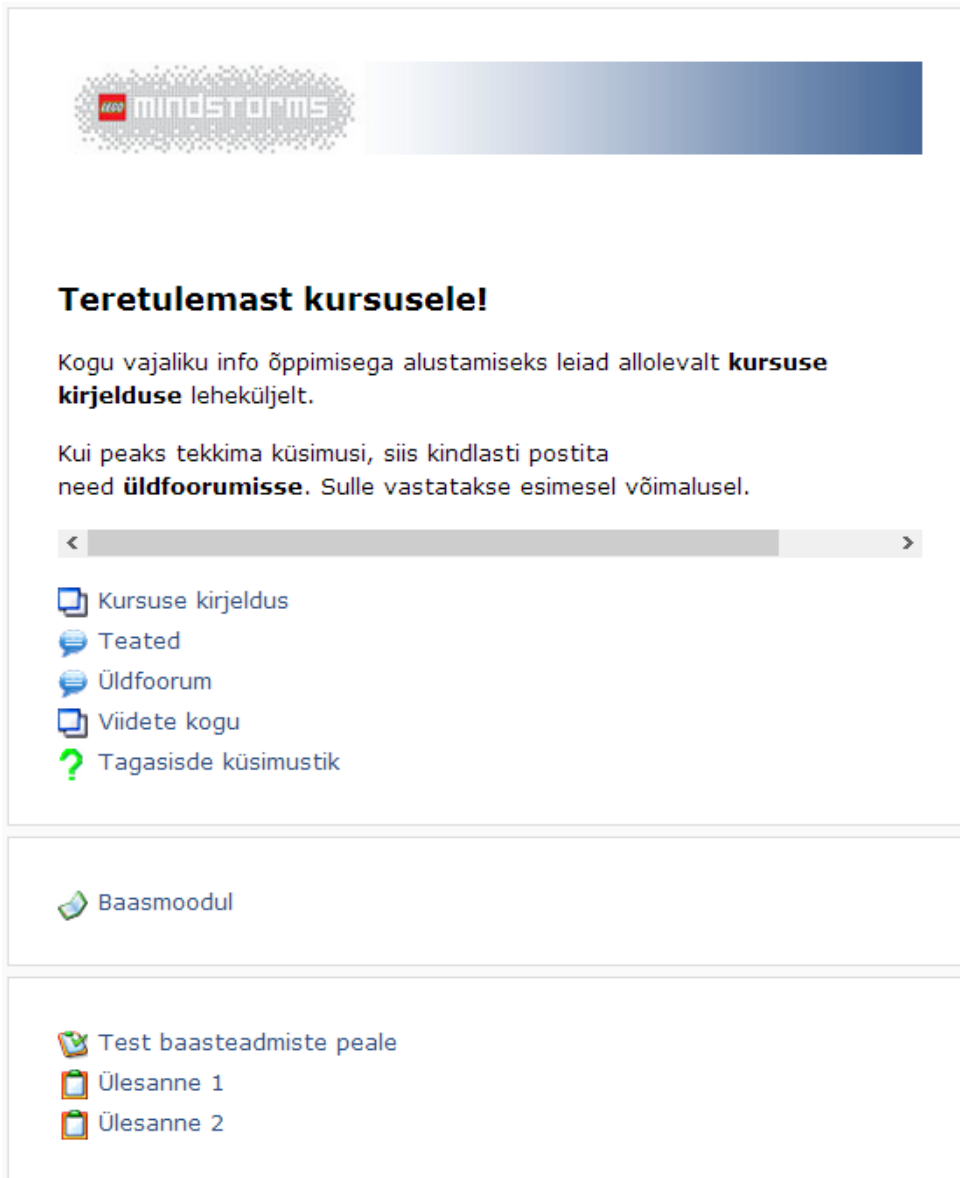
Hästi koostatud tekst peaks järgima järgnevaid põhimõtteid [40]:

- Pealkiri kajastab tekstis räägitavat
- Sissejuhatav lõik võtab kokku teksti põhisisu ja mõtte
- alapealkirjade abil jaotatakse tekst osadeks, et õppija saaks sellest parema ülevaate
- lõpuosa annab õppijale kokkuvõtte põhisisust ning võimaluse edasi liikumiseks
- pildid ja joonised visualiseerivad edastatud teavet

Samuti kasutatakse teksti kujundamisel pigem tagasihoidlikku värvigammat ning sarnaste elementide kujundus on sama.

Tuginedes Nils Austa ja Lehti Pildi e-kursuse “Videote kasutamine Moodle kursusel” [49] käigus loodud erinevate videote esitlusvahendite võrdlustabelile, otustati, et kõige parem on kasutada videote ettemängimiseks YouTube.com keskkonna võimalusi [50, 51]. Selles keskkonnas asuvat videot on lihtne lisada kursuse materjalidesse ning YouTube.com keskkonda on võimalik üles laadida kõiki enimlevinud videote formaate. Seega ei ole vaja eelnevalt tegeleda videote konverteerimise protsessiga.

Joonisel 16 on kujutatud loodava e-kursuse kavandatud väljanägemine. Lähtutud on olemasoleva LEGO Mindstorms NXT algkursuse disainist ja ülesehitusest, mille kujundajaks ja loojaks oli Rudolf Elbrecht oma magistritöö “Kvaliteetse mitmetasemelise e-kursuse loomine robotika näitel” raames [51]. Kursuse pealehel asuvad viited Kursuse kirjeldusele, Üldfoorumile, Viidete kogule, Baasmoodulile, Testile baasteadmiste peale ning ülesannetele.



**LEGO mindstorms**

## Teretulemast kursusele!

Kogu vajaliku info õppimisega alustamiseks leiad allolevalt **kursuse kirjelduse** leheküljelt.

Kui peaks tekkima küsimusi, siis kindlasti postita need **üldfoorumisse**. Sulle vastatakse esimesel võimalusel.

< >

- Kursuse kirjeldus
- Teated
- Üldfoorum
- Viidete kogu
- Tagasiside küsimustik

Baasmoodul

- Test baasteadmiste peale
- Ülesanne 1
- Ülesanne 2

**Joonis 16** E-kursuse “LEGO Mindstorms NXT roboti programmeerimine keeles NXC” pealeht

**Kursuse kirjelduse** all on kirjas info kursuse läbimise tingimuste, õppematerjalide, õppeprotsessi kirjelduse ja ajakava kohta ning moodulite, ülesannete ja nõuete kirjeldus (õpijuhised). Arvestama peab sellega, et e-kursusel õppija tegeleb õppimisega töö ja pere kõrvalt. Seega peaks õpijuhised aitama õppijal kursusega järje peale saada ka siis, kui õppija mingite olude sunnil ei saa mõnda aega kursusel osaleda. Kui kursust hakatakse läbi viima, siis tuleks sinna juurde lisada olulised reeglid ja tähtajad. Täpsema ülevaate loodud kursuse kirjeldusest saab magistritöö lisana kaasasolevalt CD plaadilt (vaata Lisa 1).

**Teated** on uudiste tüüpi foorum, mis on mõeldud teadete lihtsaks ning efektiivseks edastamiseks. See luuakse igale uuele kursusele automaatselt ning sellesse uue teema lisamise õigus on ainult õppejõul. Uusi postitatud teateid on võimalik näha ka värskete uudiste all kursuse pealehel üleval paremas servas. Nii saab kursusele siseneja kohe vaadata, kas õppejõud on edastanud mingeid uusi teateid [51 lk 53].

**Üldfoorum** on mõeldud kursusel õppijate omavaheliseks suhtluseks ning küsimuste esitamiseks õppejõule. Sinna on võimalik kõigil osalejatel luua uusi aruteluteemasid.

**Viidete kogu** sisaldab endas viiteid teistele lisamaterjalidele, mis ei kuulu kursuse kohustuslike materjalide hulka. Lisalugemise eesmärgil neile, kes soovivad antud teemaga rohkem süvitsi minna [51 lk 55].

**Tagasiside küsimustik** loomisel Moodles on kasutatud juba teiste kasutajate poolt loodud avalikke malle. Mallides leiduvad küsimused saab enda küsimustikule lisada ning seejärel neid muuta ja kustutada [51 lk 52]. Tagasiside küsimustik on e-kursuse oluline osa, kuna see aitab saadud tagasiside põhjal kursust paremaks muuta. Küsimustik koosneb 16 küsimusest, millest 12 on kohustuslikud. See hõlmab endas küsimusi kursuse struktuuri, kujunduse, sisu, õpiülesannete, läbiviimise ja hindamise kohta. Soovituslike küsimustega on küsitud, mis antud kursuse juures meeldis, mis ei meeldinud, mis oleks võinud olla teisiti ning milliseid peamisi probleeme kursuse jooksul kogesid.

**Baasmooduli** alla on lisatud kursuse kohustuslikud õppematerjalid. See jaguneb neljaks suuremaks osaks: sissejuhatus, olemasolevad vahendid NXC kasutamiseks, NXC keel ja ülesanded. Esimene osa sisaldab endas taustainfot, NXT-G ja NXC keele võrdlust, tuuakse välja NXC keele eelised ning nimetatakse NXC keele arenduskeskkondi. Teine osa keskendub NXC arenduskeskkondade kasutus- ning installeerimisjuhenditele. Kolmandas osas antakse ülevaadet NXC keele leksikaalsest ehitusest, struktuurist, operaatoritest, juhtstruktuuridest, preprotsessori käskudest ja NXC programmi käivitamise etappidest. Lisaks on loodud õppematerjali vahepeal näiteid ning vaheülesandeid teadmiste paremaks omandamiseks. Neljas osa sisaldab endas

erineva raskusastmega ülesandeid. Täpsema ülevaate loodud õppematerjalidest ja ülesannetest saab magistritöö lisana kaasasolevalt CD plaadilt ( vaata Lisa 2).

**Test baasteadmiste peale** sisaldab endas küsimusi, mis on koostatud baasmaterjali põhjal. Test tuleb sooritada pärast baasmaterjali omandamist. Test sisaldab järgnevaid küsimusi: valikvastused, õige/vale, lühivastus, vastavusse seadmine ja lünktekst. Täpsemalt testi seadetest tuleb juttu alapeatükis 3.4.

**Ülesanne 1** ja **Ülesanne 2** sisaldavad endas failide üleslaadimise võimalust. Kursuse jooksul tuleb esitada hindamiseks kaks lahendatud ülesannet, mis tuleb valida baasmooduli ülesannete peatüki alt. Ülesanne 1 tuleb valida **Ülesanded algajatele** ploki hulgast ja ülesanne 2 **Ülesanded edasijõudnutele** ploki hulgast. Alternatiivina võib teise ülesandena (**Ülesanne 2**) esitada ka mõne enda väljamõeldud huvitava ülesande koos lahendusega.

Antud peatükk andis ülevaate kursuse loomise protsessist ning tutvustas uue e-kursuse ülesehitust.

### 3.4 Test baasteadmsite peale

Ühe e-kursuse osana tuli luua hinnatav test, mis kontrolliks õppijate omandatud teadmisi. Moodle testide koostamise vahend võimaldab koostada teste 10-st eri tüüpi küsimustest nii õppijatele enesekontrolliks kui ka arvestuslikuks hindamiseks. Küsimusi (va esseeküsimus) on arvuti võimeline automaatselt kontrollima ja hindama. Testi pole võimalik sooritada anonüümselt. Kõik tulemused salvestatakse ja on õppejõule nähtavad [52]. Täpsem info testi läbimise tingimuste kohta asub testi kirjelduses.

Ajastamise seadete alt on võimalik määrata testi avanemise ja sulgemise ajad. Kui neid aegu mitte määrata, siis on test avatud kogu kursuse vältel. Ajalimiit tähistab aega, mille jooksul peab õppija testi sooritama. Ajalimiidiks on pandud 30 minutit, mis peaks olema piisav aeg testi sooritamiseks. Küsimusi on kokku 20 ning neid näidatakse 5 kaupa ühel lehel. Lubatud katsete

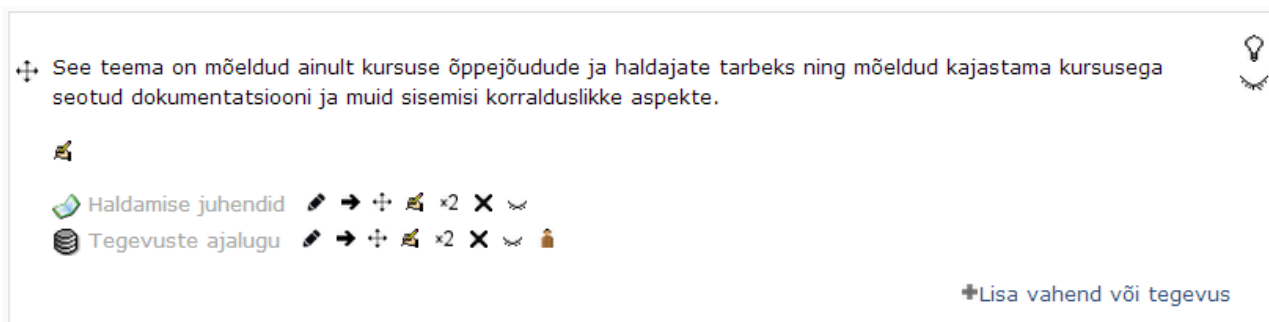
arvuks on kaks ning ajavaheks esimese ning teise katse vahel on määratud kaks päeva. See aeg on mõeldud õppimiseks ning uue katse ettevalmistuseks.

Küsimused esitatakse juhuslikus järjekorras ning hindamise meetodiks on parima katse tulemus. Küsimused on lisatud testi küsimuste pangast juhuslikkuse alusel. Küsimuste panka loodi töö autori poolt eelnevalt antud teema kohta küsimused. Parema ülevaate saamiseks paigutati need kategooriasse “Vaikimisi test baastadmiste peale”. Küsimusi on võimalik sinna kategooriasse ka juurde lisada. Kõigi küsimuste puhul tuleb sisestada küsimuse nimi ning tekst. Küsimuse tekstiga koos saab näidata ka näiteks pilti.

Peatükk tutvustas Moodle e-kursusele loodud testi baastadmsite omandamise kontrolliks.

### 3.5 Kursuse haldamine

Iga e-kursuse loomisel on soovituslik dokumenteerida selle sisu, et edaspidi teistel haldajatel oleks kergem muudatusi või täiendusi teha. Kursuse haldamise lihtsustamiseks paigutati e-kursusele lisavahend, mis on nähtav ainult kursuse haldajatele (vaata joonis 17). Lähtutud on Rudolf Elbrechti loodud kursuse haldamise juhendist [51 lk 66-68].




**Joonis 17** E-kursuse haldusvahend

Moodul **Haldamise juhendid** sisaldab endas järgnevaid juhendeid:

1. Kursuse failihaldus
2. E-kursuse muutmine
3. Õppematerjalide muutmine
4. Testi muutmine

## 5. Täiendav abi

Moodul **Tegevuste ajalugu** on andmebaasi tüüpi ning see on mõeldud suuremate muudatuste dokumenteerimiseks. See sisaldab endas nelja välja: **kuupäev**, **teostaja nimi**, **muudatuse koht** ja **muudatuse sisu**. Pärast iga tehtud suuremat muudatust tuleks sinna sisestada kirje (vaata joonis 18).

Kuupäev:	21 märts 2013
Teostaja nimi:	Arvi Kiik
Muudatuse koht:	III peatükk
Muudatuse sisu:	Sisu loomine.
	

**Joonis 18** Näide baasi sisestatud kirjest

See võimaldab saada ülevaadlikku infot e-kursusel tehtud muudatustest.

Järgnevalt on täpsemalt lahti seletatud kursuse haldamise erinevad osad.

### Kursuse failihaldus

Kursuse failid on kättesaadavad **Seadistuste** ploki **Kursuse administreerimise** valiku alt. Kõik kursusega seotud failid on sinna üles laetud (vaata joonis 19). Need on vastavalt kursuse osadele jaotatud kaustadeks. Näiteks Baasmoodul sisaldab endas faile, mida kasutatakse baasmooduli õppematerjalides. Kui peatükk sisaldab endas alapeatükke, siis nende puhul on vastavasse peakataloogi loodud peatüki nimega alamkaust.



Kursuse failid?			
Nimi	Viimati muudetud	Suurus	Tüüp
 Baasmoodul	21.03.2013 13:58		
 Haldamise juhend	21.03.2013 13:55		
 main	6.02.2013 16:28		
<a href="#">Muuda kursuse faile</a>			

### Joonis 19 Kursuse kataloogistruktuur

Tekstiredaktoris näiteks uue pildi lisamisel tuleks see kõigepealt oma arvutist üles laadida kursuse failide hulka lähtudes olemasolevast kataloogistruktuurist ning seejärel tekstiredaktoris see kursuse failide kataloogist üles otsida.

#### E-kursuse muutmine

E-kursusel muudatuste tegemiseks tuleb kursuse pealehe paremas ülaservas klikkida nupul “**Luba muutmine**”. Uute vahendite ja tegevuste lisamine toimub välja “**Lisa vahend või tegevus**” alt. Iga kursusele lisatud vahendi ja tegevuse järel on rida nuppe. Vahendite ja tegevuste muutmiseks tuleb klikata olemasoleva vahendi taga asuvale muutmist lubavale ikoonile. Seal asub ka kustutamise ikoon. Oluline on meeles pidada, et kord kustutatud vahendit või tegevust pole võimalik taastada. Täpsema kirjelduse kõigist vahendite seadistuste nuppudest leiab Moodle juhendist “Kuidas alustada uue e-kursuse loomisega Moodle’is” alapeatüki Vahendite ja tegevuste muutmine alt [43].

#### Õppematerjalide muutmine

Õppematerjalide muutmiseks tuleb klikkida pealehel baasmooduli peal. Seejärel tuleb valida seadistuste paneelilt “**Luba muutmine**”. Vasakul sisukorra paneelil ilmub selle tulemusena iga peatüki taha terve hulk ikoone. Klikkida tuleb “**Muuda objekti**” ikoonil. Ette kuvatakse tekstitoimetuste aken, kus on olemas üldtuntud nupud teksti kujundamiseks, piltide, tabelite ja veebilinkide lisamiseks. Detailsema juhendi koos piltidega leiab juhendist “Õppematerjalide loomine ja lisamine Moodles” alapeatüki “Õppematerjalide loomine Moodles” alt [47].

### **Testi muutmine**

Test haldamiseks tuleb klikkida testi nime peal ning seejärel ilmub vasakule seadistuste plokk, mis sisaldab endas valikuid testi haldamiseks. “**Muuda testi**” valiku alt on võimalik testist eemaldada ning testi lisada uusi küsimusi. Testile saab lisada küsimusi, mis tuleb eelnevalt luua küsimuste panka. Küsimuste panka on võimalik luua ka uus kategooria küsimuste paremaks eristamiseks. Hetkel lisatakse testi juhuslikud küsimused küsimuste panga kategooriast “**Vaikimisi test baasteadmiste peale**”.

Täpsemalt kõigist testi haldamise seadetest ning erinevate tüüpi küsimuste loomisest ja lisamisest küsimuste panka saab hea ülevaate “Testide koostamine Moodle’is” juhendist [52].

### **Täiendav abi**

Täiendava abi saamiseks on mõttekas tutvuda ka Tartu Ülikooli kodulehel asuvate kõikide teiste Moodle juhenditega [53, 54] või pöörduda mõne Tartu Ülikooli haridustehnoloogi poole [55].

Täpsema ülevaate loodud kursuse haldamise juhendist saab magistritöö lisana kaasasolevalt CD plaadilt ( vaata Lisa 3 ).

## Kokkuvõte

Käesoleva magistritöö peamiseks eesmärgiks oli luua kvaliteetne Moodle e-kursus RAJU keskuse poolt korraldavatele õpetajakoolitustele, millest võtavad osa robotika õpetajad. Enne kursuse loomist analüüsiti uue kursuse loomise vajalikkust ning toodi välja põhjuseid, miks on vaja üle minna kõrgemates klassides LEGO Mindstorms NXT roboti programmeerimiseks kasutatava graafilise programmeerimiskeele NXT-G pealt tekstipõhise programmeerimiskeele NXC peale.

Töö jaguneb kolmeks osaks. Esimeses osas tutvustatakse tuntumaid olemasolevaid programmeerimiskeeli, mida saab kasutada LEGO Mindstorms NXT roboti programmeerimiseks ning seejärel põhjendatakse, miks peaks teiseks programmeerimiskeeleks pärast NXT-G keele selgeksõppimist olema NXC keel. Tutvustatakse põgusalt NXC programmi loomist ja selle käivitamise etappe, muutujate tüüpe, aritmeetilisi operaatoreid ning täiendatud omistustehteid. Võrreldakse NXT-G ja NXC keelest baitkoodi teisendatud sama funktsiooni täitvate programmide mälukasutust ja kiirust ning tuuakse välja saadud tulemused.

Teises osas tutvustatakse Priit Ranna loodud programmi NXCEesti. Rakendus töötab nii Windows, Linux kui ka Mac OS operatsioonisüsteemidel ning võimaldab kasutada põhilisi tekstiredaktori võimalusi. Igal platvormil on oma kompilaator, mille kasutamise eest vastutab programm. Kõigepealt kirjeldatakse, kuidas programmi paigaldada, seejärel tuleb juttu programmi seadistamise võimalustest ning antakse ülevaade kasutajaliidest. Programm võimaldab kasutada eestikeelseid funktsioone ja tõlkeid. Seletatakse, mis põhimõttel on need tõlked loodud ning tuuakse näiteid tõlgete kasutamise võimalustest. Antud programmi kasutamise võimalusi koos tõlgetega õpetatakse ka loodud Moodle kursusel.

Töö kolmas osa tutvustab magistritöö käigus valminud e-kursuse loomise protsessi. Alustatakse analüüsiga, mille käigus lahutatakse terviklik teema lihtsamateks osadeks. Toimub vajaduste, sihtrühma ja konteksti analüüs. Selgitatakse välja kursuse nõuded ja õpieesmärgid. Seejärel tutvustatakse kursuse ajakava, hindamiskriteeriume ja tegevusi ning kursuse loomise käiku õpikeskkonnas Moodle. Määratakse nõuded õppematerjalile ning kirjeldatakse kursuse erinevaid osi. Lõpus antakse ülevaade ka loodud kursuse haldamise võimalustest.

Kokkuvõttes võib öelda, et uue kvaliteetse e-kursuse loomise protsess on väga ajamahukas ning nõuab põhjalikku analüüsi, ettevalmistust ning planeerimist. Eriti kui õppematerjale pole valmiskujul olemas ning need tuleb endal luua. Seetõttu tuleks kindlasti jälgida, et õppematerjalid, mida luuakse, oleks korduvkasutatavad. Antud töö edasiarendusena näeks töö autor baasmooduli kõrvale teiste õppemoodulite lisamist (näiteks eraldi NXT andurite moodul või ülesannete moodul).

# **LEGO Mindstorms NXT robotics programming in NXC language**

## **(e-learning course creation)**

**Master's Thesis (30EAP)**

**Arvi Kiik**

## **Summary**

The main purpose of this master thesis was to create high-quality e-learning robotics course for robotics teachers. This course teach how to program LEGO Mindstorms NXT robot using NXC language which is text based programming language. During the creation of the course all e-learning materials had to be created too.

Within the thesis an extensive analysis is done. The analysis shows that there is a need for new e-learning course. The course was created in the Moodle learning management system because it has greates tools for structuring the course content (upload files, learning modules, media library), for communication (forum, chat), for learning (tasks, tests, group work equipment) and for course management. During the creation process of the new e-learning course it was taken into account e-learning quality requirements.[40]

This thesis consist of three parts. The first part introduces the languages that can be used to program the LEGO Mindstorms NXT robot and gives reasons why it should be teach new programming language after the NXT-G language which is a graphical programming language. Also the first part introduces NXC program start-up phases, types of variables, arithmetic operators, compares the speed and memory usage of the NXC and NXT-G program, which fulfill the same function and are converted into NXT byte code. The second part provides an overview of the NXCEesti program, which is created by Priit Rand. First of all, it is described how to install the program, then it is given an overview of the user interface configuration options. Also it is shown how to use estonian functions, translations and how those translations are created. The third part concentrates on the process of creating a new e-learning course. It is described

course requirements and learning objectives. This part gives an overview the parts of the course (course decription, formums, tasks, test, learning materials and the course management guide).

# Kasutatud kirjandus

- 1 M. Dremljuga-Telk, E. Koitla, K. Kusnets, M. Kusmin, M. Niitsoo, L. Pilt, T. Plank, M. Varendi, A. Villemis, "E-learning Quality Assurance System for e-Courses in Estonia," lk 1-4 <http://www.e-oee.ee/download/repository/TEPE.pdf> (18.04.2013)
- 2 S. Hassenplug, "NXT Programming Software," 29.02.2008, <http://www.teamhassenplug.org/NXT/NXTSoftware.html> (18.04.2013)
- 3 "EDU NXT tarkvara," [http://www.robomiku.ee/pood/product.php?id\\_product=64](http://www.robomiku.ee/pood/product.php?id_product=64) (18.04.2013)
- 4 J. Hannsen, "NXT Power Programming - book on NXC," 11.10.2007 <http://thenxtstep.blogspot.com/2007/10/nxt-power-programming-book-on-nxc.html> (18.04.2013)
- 5 "Bricx Command Center 3.3," <http://bricxcc.sourceforge.net/> (18.04.2013)
- 6 "Programmable Brick Utilities," <http://bricxcc.sourceforge.net/utilities.html> (18.04.2013)
- 7 D. Benedettelli, "Programming LEGO NXT Robots using NXC," lk 8, 07.06.2007, [http://bricxcc.sourceforge.net/nbc/nxcdoc/NXC\\_tutorial.pdf](http://bricxcc.sourceforge.net/nbc/nxcdoc/NXC_tutorial.pdf) (18.04.2013)
- 8 "RobotC for Mindstorms," [http://www.robotc.net/support/nxt/MindstormsWebHelp/index.htm#page=getting\\_started/sampleprograms/Sample%20Programs.htm](http://www.robotc.net/support/nxt/MindstormsWebHelp/index.htm#page=getting_started/sampleprograms/Sample%20Programs.htm) (18.04.2013)
- 9 "Robolab," <http://ceeo.tufts.edu/Products-Robolab/robolab.html> (18.04.2013)
- 10 "LabView add-ons for LEGO Mindstorms NXT," <http://www.ni.com/white-paper/4435/en> (18.04.2013)
- 11 "The LeJOS NXJ Tutorial," <http://lejos.sourceforge.net/nxt/nxj/tutorial/index.htm> (18.04.2013)
- 12 "Python programming language," [http://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Python_(programming_language)) (18.04.2013)
- 13 "Object-oriented programming," [http://en.wikipedia.org/wiki/Object-oriented\\_programming](http://en.wikipedia.org/wiki/Object-oriented_programming) (18.04.2013)
- 14 "Procedural programming," [http://en.wikipedia.org/wiki/Procedural\\_programming](http://en.wikipedia.org/wiki/Procedural_programming) (18.04.2013)

- 15 “Functional programming,” [http://en.wikipedia.org/wiki/Functional\\_programming](http://en.wikipedia.org/wiki/Functional_programming) (18.04.2013)
- 16 “NXT-python,” <http://code.google.com/p/nxt-python/>
- 17 D. P. Lau, “NXT\_Python,” [http://home.comcast.net/~dplau/nxt\\_python/](http://home.comcast.net/~dplau/nxt_python/) (18.04.2013)
- 18 “Welcome to Next Byte Codes, Not eXactly C and SuperPro C,”  
<http://bricxcc.sourceforge.net/nbc/> (18.04.2013)
- 19 A. Lauringson, “Tartus peetakse Eesti koolirobootika suurfinaali,”  
<http://www.tiigrihype.ee/et/uudised/tartus-peetakse-eesti-koolirobootika-suurfinaali>
- 20 “LEGO Mindstorms NXT 2.0,”  
[http://en.wikipedia.org/wiki/Lego\\_Mindstorms\\_NXT\\_2.0](http://en.wikipedia.org/wiki/Lego_Mindstorms_NXT_2.0) (18.04.2013)
- 21 H. Andersson, “LEGO Mindstorms Sudoku Solver,”  
<http://www.youtube.com/watch?v=Mp8Y2yjV4fU&feature=related> (18.04.2013)
- 22 M. Vunk “LEGO Mindstorms NXT” ga ühilduv käsidünamomeeter, ” lk 23-27, 2012  
(18.04.2013)
- 23 “NXC Introduction,” <http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/intro.html>  
(18.04.2013)
- 24 “C sissejuhatus,” <https://et.wikibooks.org/wiki/C:Sissejuhatus> (18.04.2013)
- 25 “Robotics Programming in C,” lk 2-3, <http://personalpages.tds.net/~jwbacon/tutorial.pdf>  
(18.04.2013)
- 26 H. H. Cheng, “Ten reasons to teach and learn computer programming in C,”  
<http://iel.ucdavis.edu/publication/WhyC.html> (18.04.2013)
- 27 “TIOBE programming community index for April 2013,”  
<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> (18.04.2013)
- 28 “LeJOS NXJ getting started on Microsoft Windows,”  
<http://lejos.sourceforge.net/nxt/nxj/tutorial/Preliminaries/GettingStartedWindows.htm>  
(18.04.2013)
- 29 “The City Collegue,” <http://www.citycollege.ac.uk/Courses/Info/CP.html> (18.04.2013)
- 30 J. Hansen, “Not eXactly C programmers guide, ” lk 1-18,  
<http://lego.itam.mx/misc/manuales/NXC/guide.pdf> (18.04.2013)
- 31 E. Moorits, “Programmi kompileerimine ja ja C eripärad.  
<http://www.pld.ttu.ee/IAF0542/2011/loeng12.pdf> (18.04.2013)



- 32 “LEGO Mindstorms,” [http://et.wikipedia.org/wiki/Lego\\_Mindstorms](http://et.wikipedia.org/wiki/Lego_Mindstorms) (18.04.2013)
- 33 “Muutujad,” [http://www.kivilinn.tartu.ee/tehnika/files/2013/01/3\\_muutujad.pdf](http://www.kivilinn.tartu.ee/tehnika/files/2013/01/3_muutujad.pdf) (18.04.2013)
- 34 “C keele programmiehitusest,” <http://www.hot.ee/veiks26/cprog/cprog21.html> (18.04.2013)
- 35 A. Sacek, “LEGO 3D Milling machine - 3D printer,” <http://www.youtube.com/watch?v=pX1cO2XhMrg> (18.04.2013)
- 36 P. Rand, “NXC Platvormiülene keskkond,” magistritöö(18.04.2013)
- 37 “Java SE downloads,” <http://www.oracle.com/technetwork/java/javase/downloads/index.html> (18.04.2013)
- 38 “Fantom driver 1.1.3,” <http://mindstorms.lego.com/en-us/support/files/Driver.aspx> (18.04.2013)
- 39 “The preprocessor,” <http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/preproc.html> (18.03.2013)
- 40 E-õppe Arenduskeskus, “Juhend kvaliteetse e-kursuse loomiseks,” [http://www.e-oep.ee/\\_download/repository/Kasiraamat2.0.pdf](http://www.e-oep.ee/_download/repository/Kasiraamat2.0.pdf) (18.04.2013)
- 41 W. Horton, E-Learning by Design, 2006, lk 364
- 42 K. Hendla, “Õpiobjektide vahetamine ARIADNE ja WebCT vahel,” lk 12, [http://www.e-oep.ee/images/50000902/KadriHendla\\_AriadneWebCT.pdf](http://www.e-oep.ee/images/50000902/KadriHendla_AriadneWebCT.pdf) (18.04.2013)
- 43 T. Marandi, “Kuidas alustada uue e-kursuse loomisega Moodle’is,” 28.09.2012, <https://wiki.ut.ee/pages/viewpage.action?pageId=17113977> (18.04.2013)
- 44 N. Konovalov, “NXC” bakalaureusetöö (17.05.2013)
- 45 “Moodle raamatu moodul,” [https://moodle.org/plugins/pluginversions.php?plugin=mod\\_book](https://moodle.org/plugins/pluginversions.php?plugin=mod_book) (18.04.2013)
- 46 T. Tartes, “Moodle installeerimine ja administreerimine,” 2010, [http://www.e-oep.ee/\\_download/euni\\_repository/file/994/materjal\\_3.pdf](http://www.e-oep.ee/_download/euni_repository/file/994/materjal_3.pdf) (18.04.2013)
- 47 T. Marandi, “Õppematerjalide lisamine ja loomine Moodle’is,” <https://wiki.ut.ee/pages/viewpage.action?pageId=17114000> (18.04.2013)
- 48 J. Metsamaa “Materjalide kooskirjutamise raamistik RAJU keskuse õppevara näitel,” lk 49, (18.04.2013)

- 49 N. Austa, L. Pilt “Videote kasutamine Moodle kursusel,”  
<https://moodle.ut.ee/course/info.php?id=645> (18.04.2013)
- 50 “YouTube keskkond,” <http://www.youtube.com/> (18.04.2013)
- 51 R. Elbrecht, “Kvaliteetse mitmetasemelise e-kursuse loomine robotika näitel”  
magistritöö (17.05.2013)
- 52 M. Piir, “Testide koostamine Moodle’is,” 04.04.2013,  
<https://wiki.ut.ee/display/moodle/8.+Testide+koostamine+Moodle%27is> (18.04.2013)
- 53 Tartu Ülikool, “Juhendid,” <http://www.ut.ee/et/372019> (18.04.2013)
- 54 L. Pilt, “Moodle’i juhendid,” 03.04.2012,  
<https://wiki.ut.ee/display/moodle/Moodle%27i+juhendid> (18.04.2013)
- 55 Tartu Ülikool, “Abi ja nõustamine,” <http://www.ut.ee/et/590929> (18.04.2013)

## **Lisad**

Antud osa all on toodud lisamaterjalid, mis on antud magistritööga kaasas. Tegemist on PDF (\*.pdf) failidega. CD plaat on kinnitatud käesoleva magistritöö tagakaane siseküljele.

### **Lisa 1**

Loodud Moodle kursuse kirjeldus, mis asub käesoleva töö lisana kaasasoleva CD plaadil Lisa 1 kaustas Kursuse kirjeldus.pdf fail.

### **Lisa 2**

Loodud Moodle kursuse õppematerjalid, mis asuvad käesoleva töö lisana kaasasoleva CD plaadil Lisa 2 kaustas Õppematerjalid.pdf fail.

### **Lisa 3**

Loodud Moodle kursuse haldamise juhend, mis asub käesoleva töö lisana kaasasoleva CD plaadil Lisa 3 kaustas Haldamise juhend.pdf fail.

# **Lihtlitsent lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks**

Mina, Arvi Kiik

(sünnikuupäev: 17.08.1988)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „LEGO Mindstorms NXT roboti programmeerimine keeles NXC (e-kursuse loomine), mille juhendajad on Anne Villemis ja Taavi Duvin,
  - 1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **22.05.2013**